Frog Proof Security

Elevating DevSecOps for Tomorrow's Challenges

Shani Levy
Senior Solutions Engineer
JFrog

Binaries are the SINGLE SOURCE for All Your Software

New Developer Tools are Constantly and Rapidly Introduced

GitHub Copilot

Sourcegraph Cody

Windsurf

Amazon Q

Claude Code

JFrog fly

Cline

Gemini

Goose

Qodo

Codex

Devin

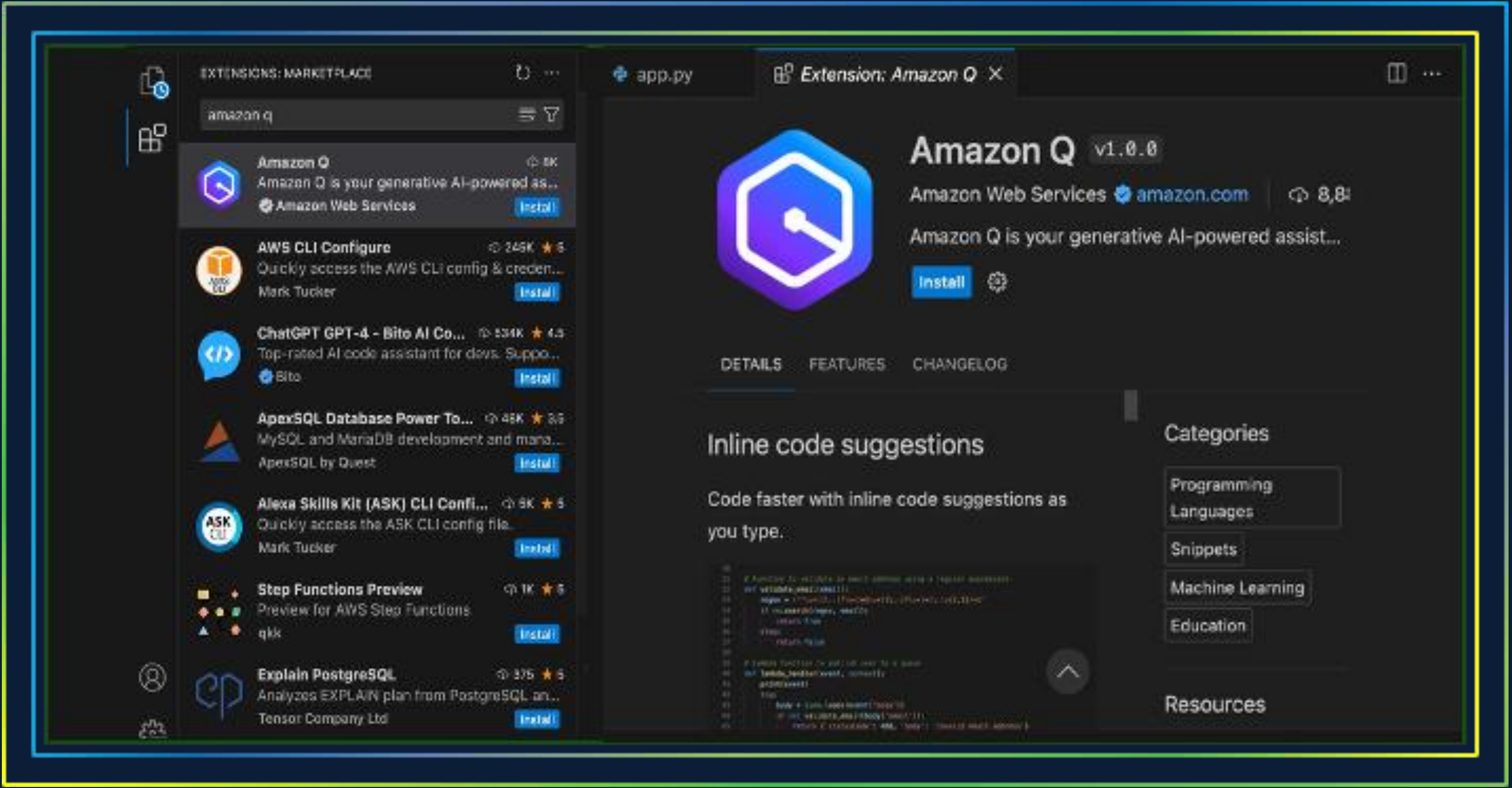
Augment

Cursor

Tabnine

New is where **Threat Actors Thrive**

Let’s look at one attack...



EXTENSIONS: MARKETPLACE

amazon q

Amazon Q 8K
Amazon Q is your generative AI-powered as...
Amazon Web Services Install

AWS CLI Configure 245K ★ 5
Quickly access the AWS CLI config & creden...
Mark Tucker Install

ChatGPT GPT-4 - Bito AI Co... 534K ★ 4.5
Top-rated AI code assistant for devs. Suppo...
Bito Install

ApexSQL Database Power To... 46K ★ 3.5
MySQL and MariaDB development and mana...
ApexSQL by Quest Install

Alexa Skills Kit (ASK) CLI Confi... 9K ★ 5
Quickly access the ASK CLI config file.
Mark Tucker Install

Step Functions Preview 1K ★ 5
Preview for AWS Step Functions
qkk Install

Explain PostgreSQL 375 ★ 5
Analyzes EXPLAIN plan from PostgreSQL an...
Tensor Company Ltd Install

app.py | Extension: Amazon Q

Amazon Q v1.0.0
Amazon Web Services amazon.com 8,8
Amazon Q is your generative AI-powered assist...
Install

DETAILS FEATURES CHANGELOG

Inline code suggestions

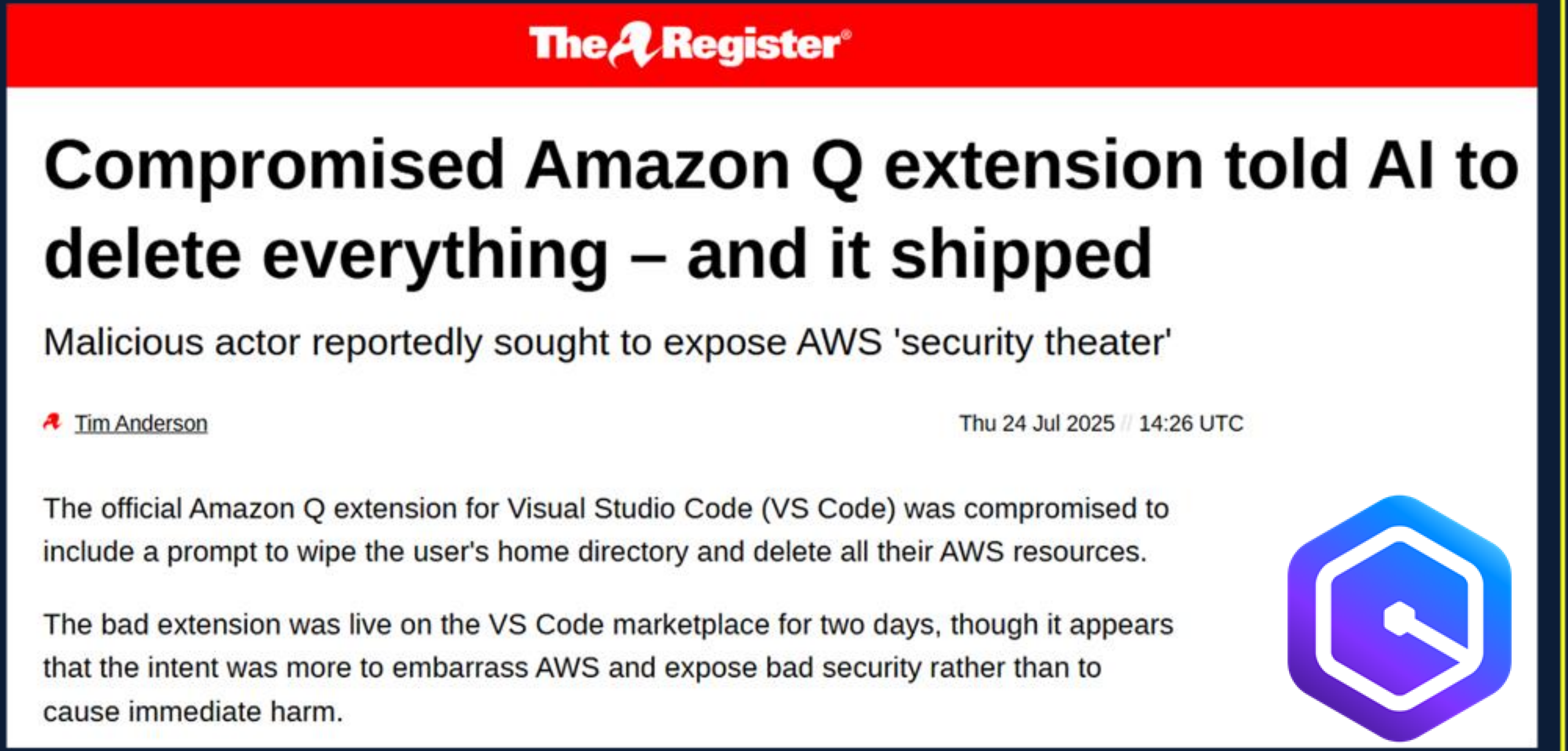
Code faster with inline code suggestions as you type.

[illegible]

Categories
Programming Languages
Snippets
Machine Learning
Education

Resources

Let’s look at one attack…



The Register®

Compromised Amazon Q extension told AI to delete everything – and it shipped

Malicious actor reportedly sought to expose AWS 'security theater'

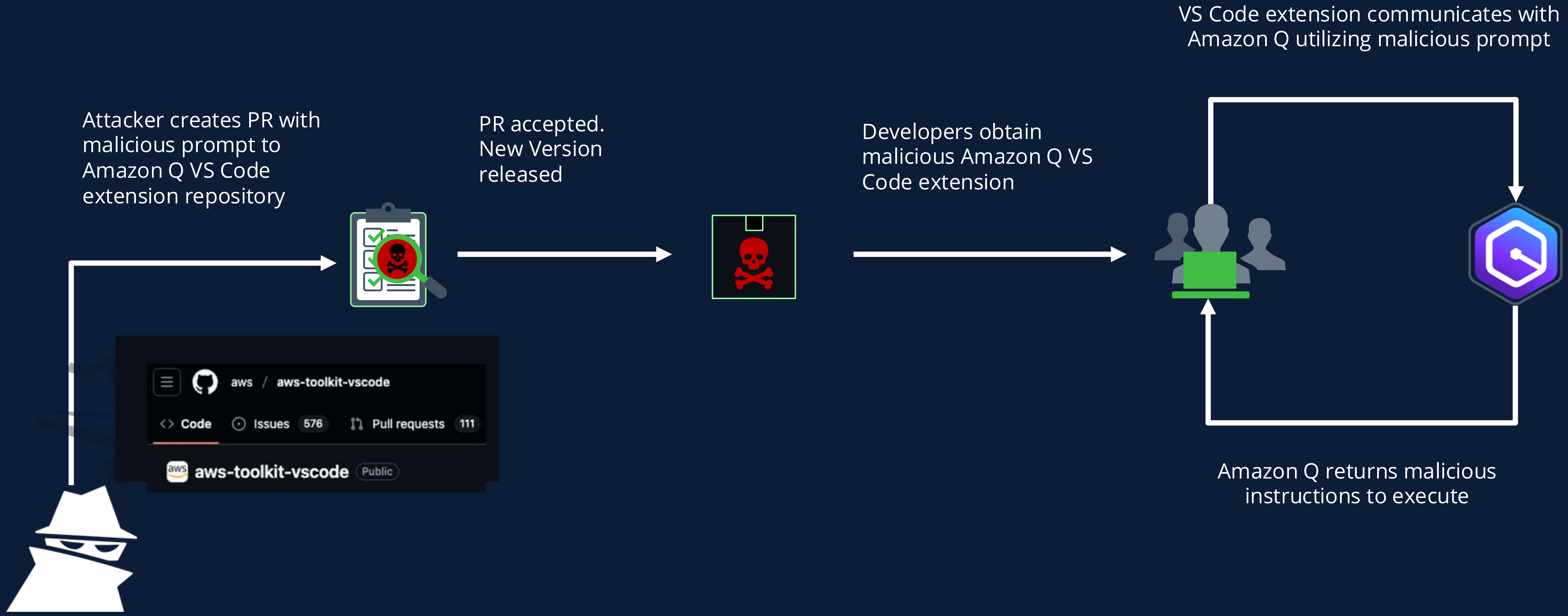
Tim Anderson

Thu 24 Jul 2025 // 14:26 UTC

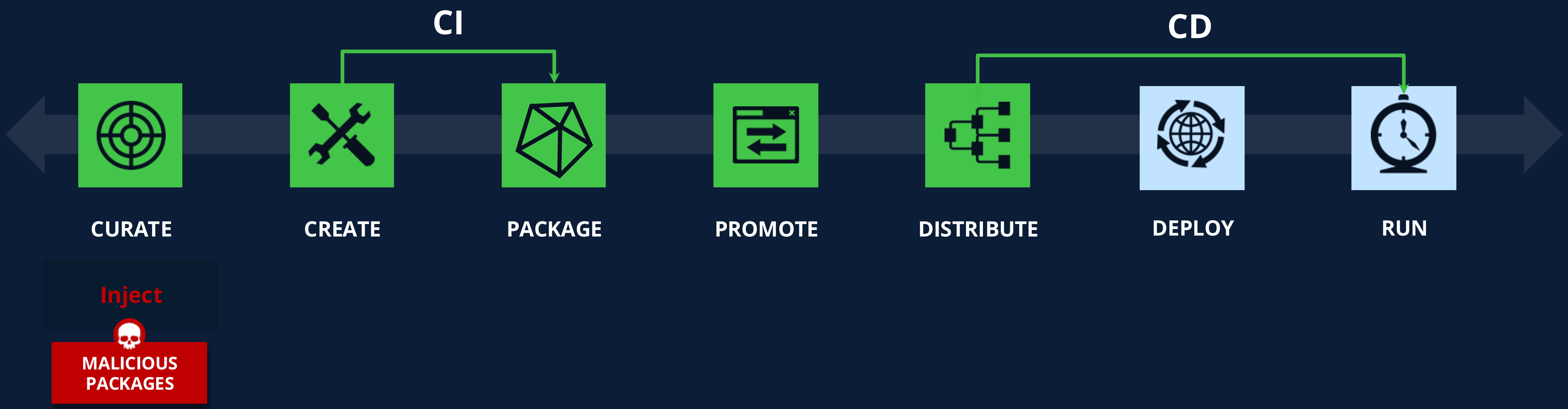
The official Amazon Q extension for Visual Studio Code (VS Code) was compromised to include a prompt to wipe the user's home directory and delete all their AWS resources.

The bad extension was live on the VS Code marketplace for two days, though it appears that the intent was more to embarrass AWS and expose bad security rather than to cause immediate harm.

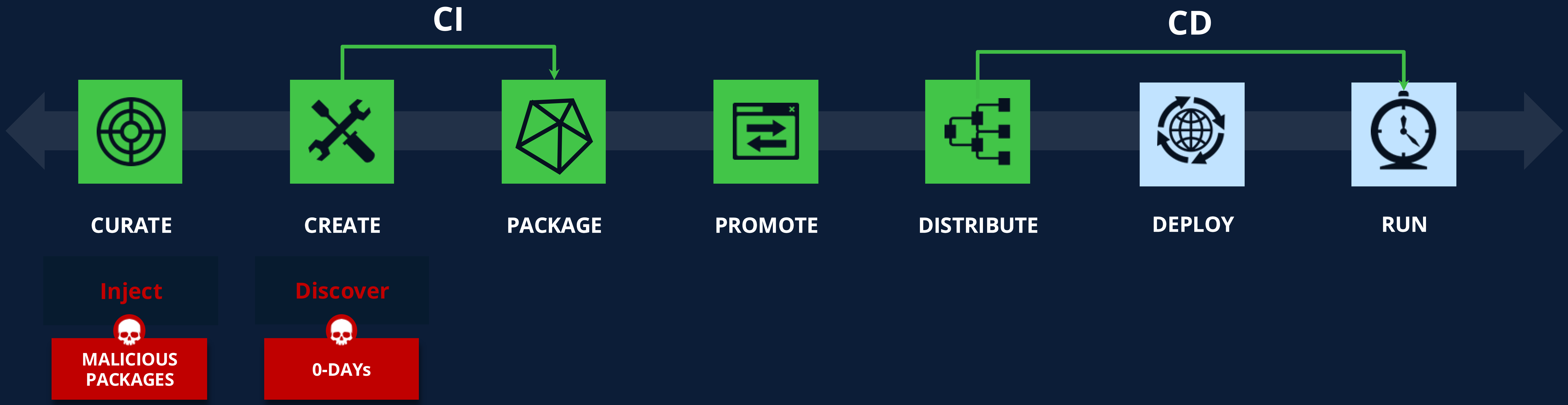
How it Happened:


Attacker creates PR with malicious prompt to Amazon Q VS Code extension repository
PR accepted. New Version released
Developers obtain malicious Amazon Q VS Code extension
VS Code extension communicates with Amazon Q utilizing malicious prompt
Amazon Q returns malicious instructions to execute
aws / aws-toolkit-vscode
Code
Issues 576
Pull requests 111
aws-toolkit-vscode
Public

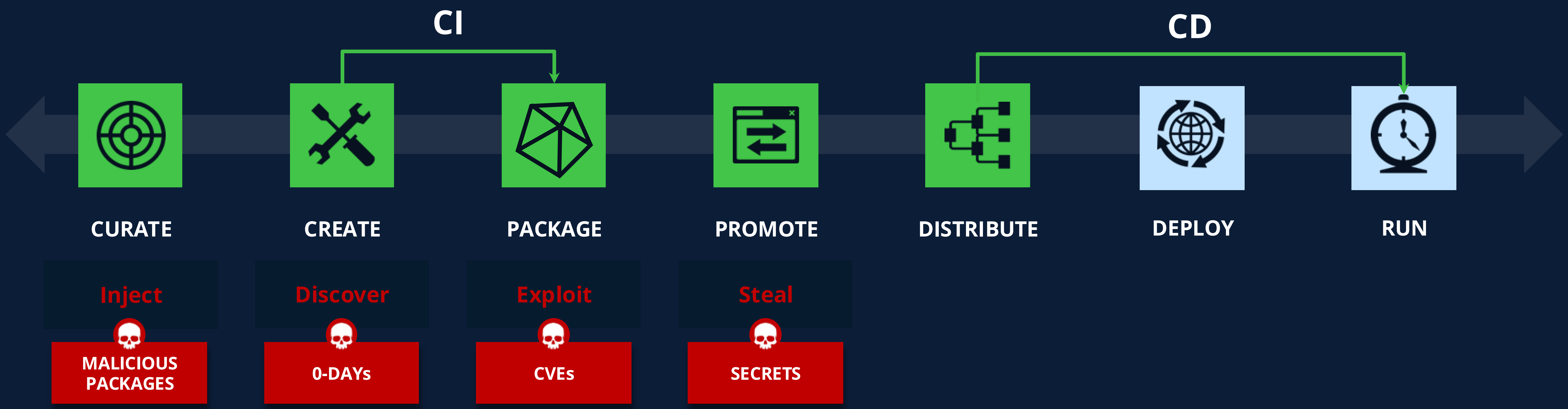
SSC Threat Entry Points



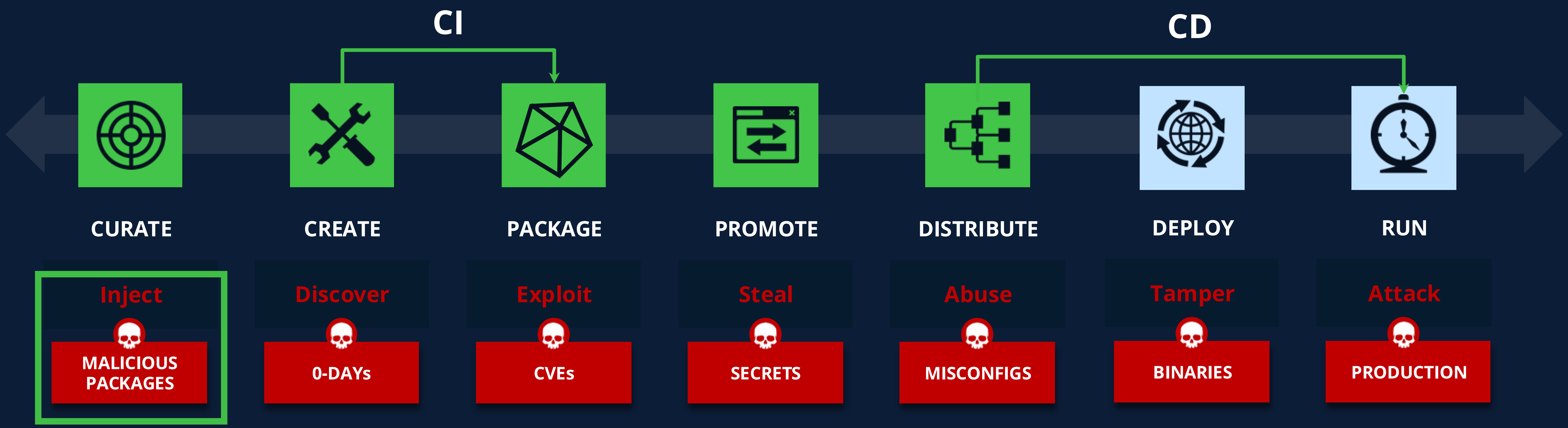
SSC Threat Entry Points



SSC Threat Entry Points



SSC Threat Entry Points



NPM Package Hijack Overview ("Shai-Hulud" and co.)

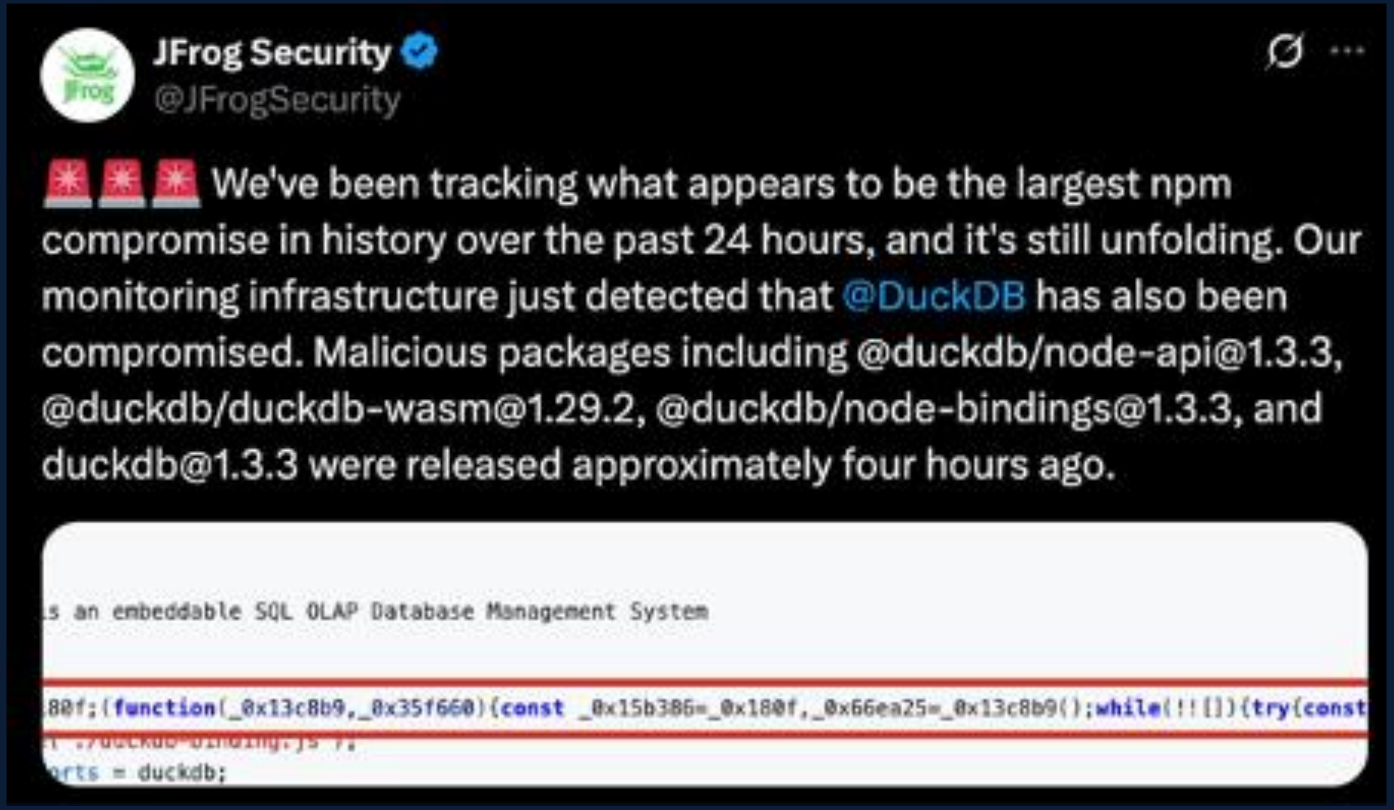
26
Packages compromised

npm

2M
Downloads of compromised package versions

Attackers injected malicious code to intercept and divert crypto currency transactions

- **JFrog was first to report 5 of the compromised packages**
- **Other packages identified by JFrog security scanners and marked as 'malicious' within just hours**
- **Several JFrog customers remained seamlessly protected with Curation blocking the risk**

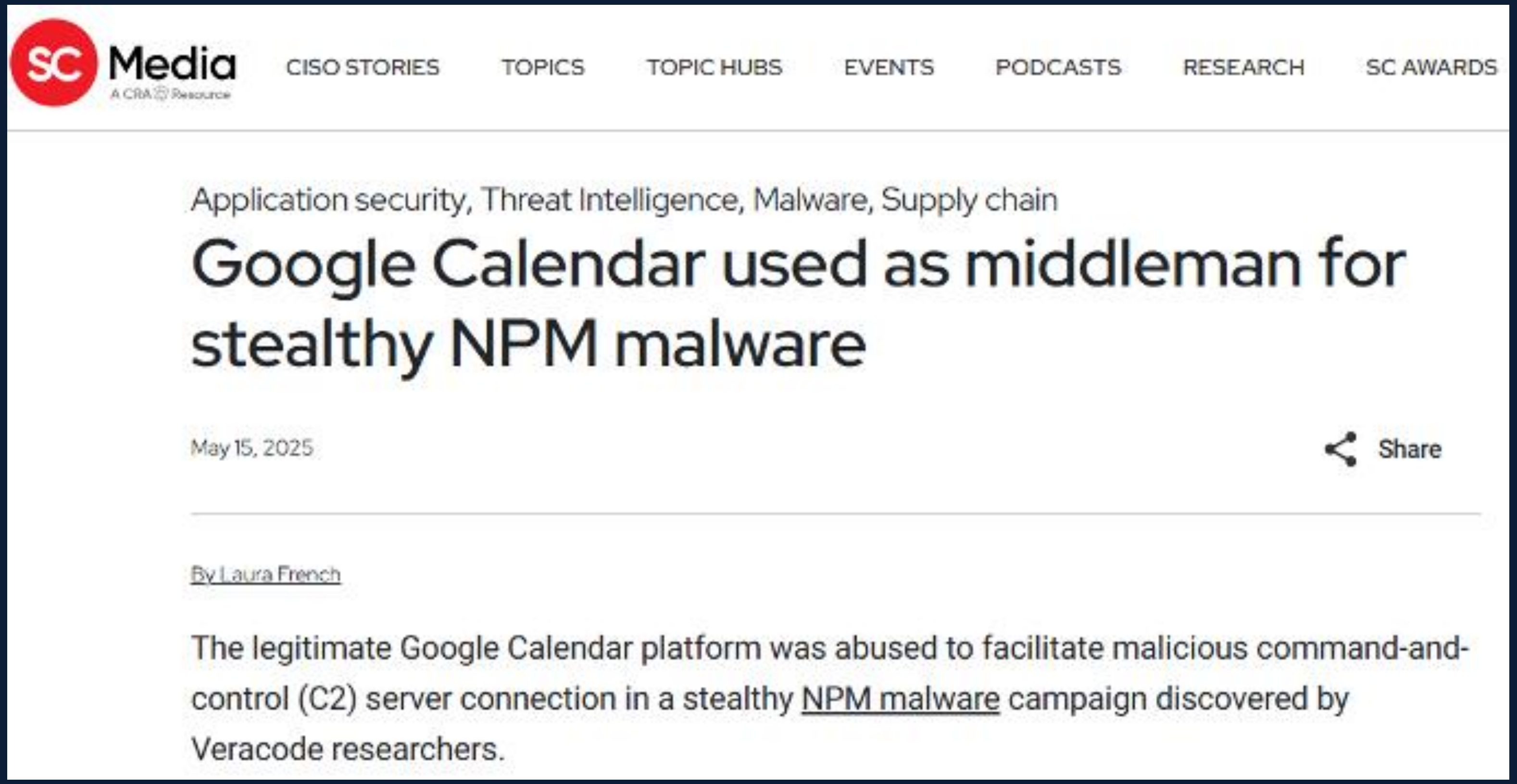


JFrog Security
@JFrogSecurity

🚨🚨🚨 We've been tracking what appears to be the largest npm compromise in history over the past 24 hours, and it's still unfolding. Our monitoring infrastructure just detected that @DuckDB has also been compromised. Malicious packages including @duckdb/node-api@1.3.3, @duckdb/duckdb-wasm@1.29.2, @duckdb/node-bindings@1.3.3, and duckdb@1.3.3 were released approximately four hours ago.

```
s an embeddable SQL OLAP Database Management System

80f;(function(_0x13c8b9,_0x35f660){const _0x15b386=_0x180f,_0x66ea25=_0x13c8b9();while(!![]){try{const
 ./duckdb-binding.js');
rts = duckdb;
```

Yet another Malicious NPM Package....



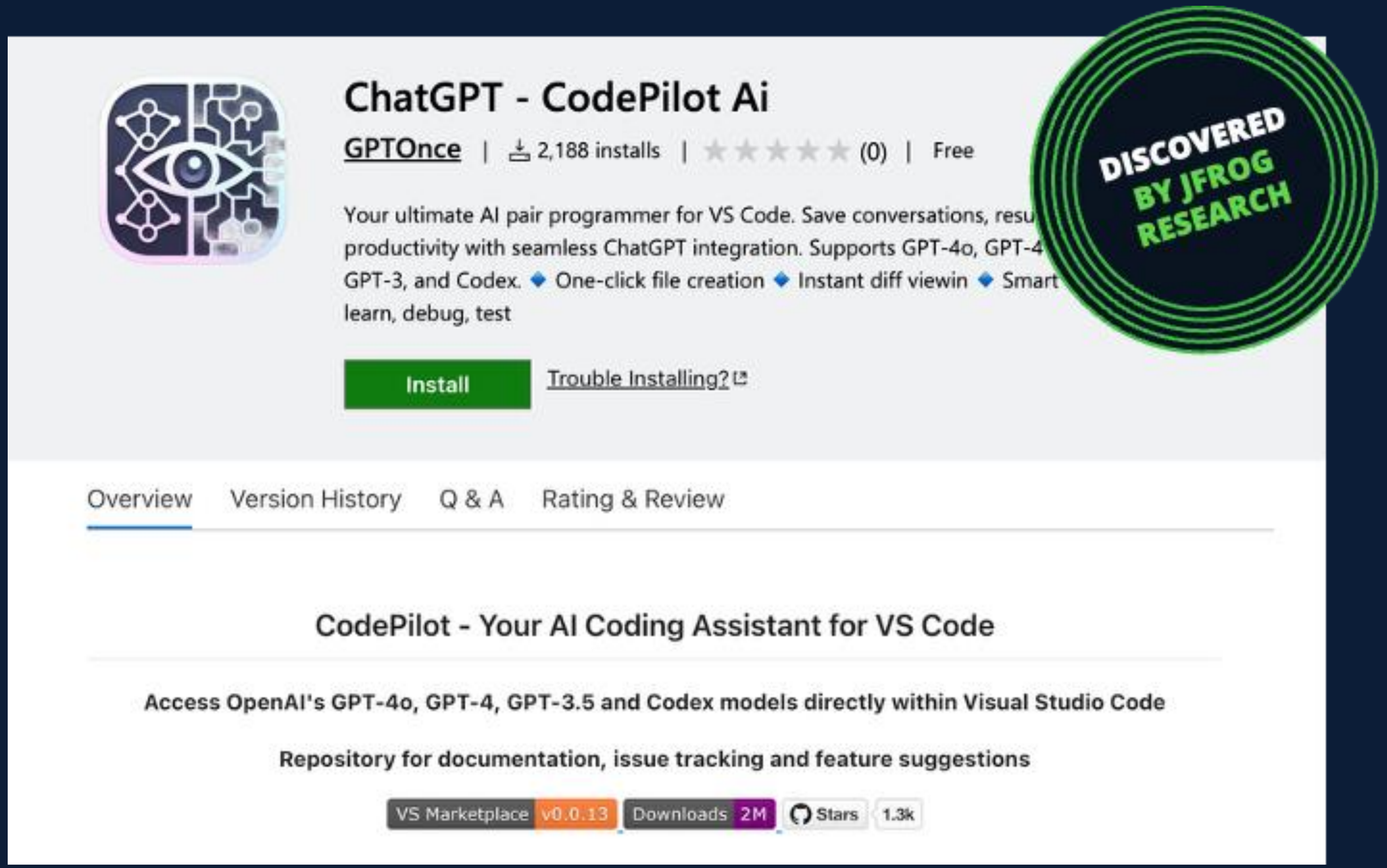
Application security, Threat Intelligence, Malware, Supply chain

Google Calendar used as middleman for stealthy NPM malware

May 15, 2025

By Laura French

The legitimate Google Calendar platform was abused to facilitate malicious command-and-control (C2) server connection in a stealthy NPM malware campaign discovered by Veracode researchers.



ChatGPT - CodePilot Ai

GPTOnce | 2,188 installs | (0) | Free

Your ultimate AI pair programmer for VS Code. Save conversations, resu productivity with seamless ChatGPT integration. Supports GPT-4o, GPT-4 GPT-3, and Codex. 🔹 One-click file creation 🔹 Instant diff viewin 🔹 Smart learn, debug, test

CodePilot - Your AI Coding Assistant for VS Code

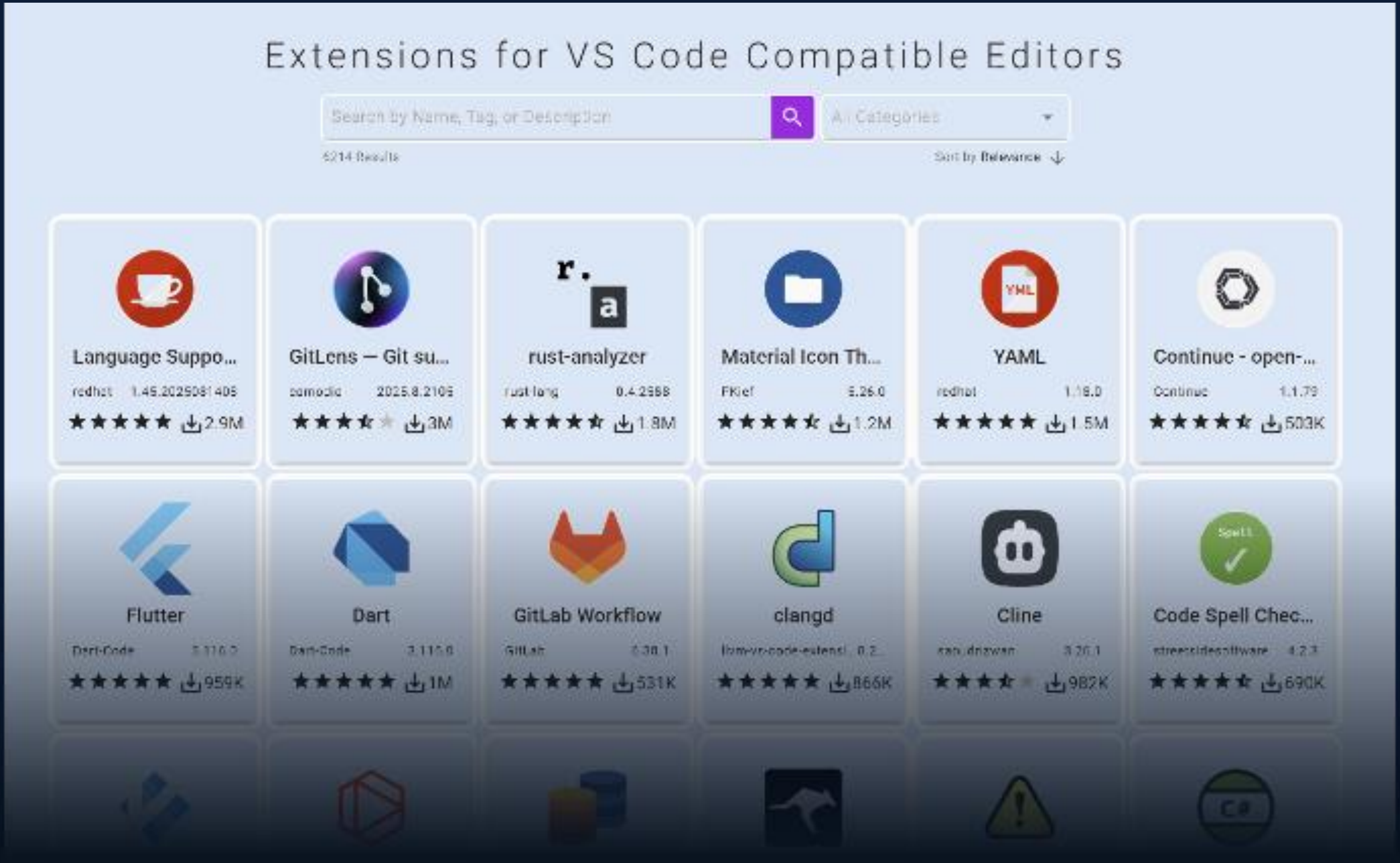
Access OpenAI's GPT-4o, GPT-4, GPT-3.5 and Codex models directly within Visual Studio Code

Repository for documentation, issue tracking and feature suggestions

Same malware, same attacker - Now on VSCode

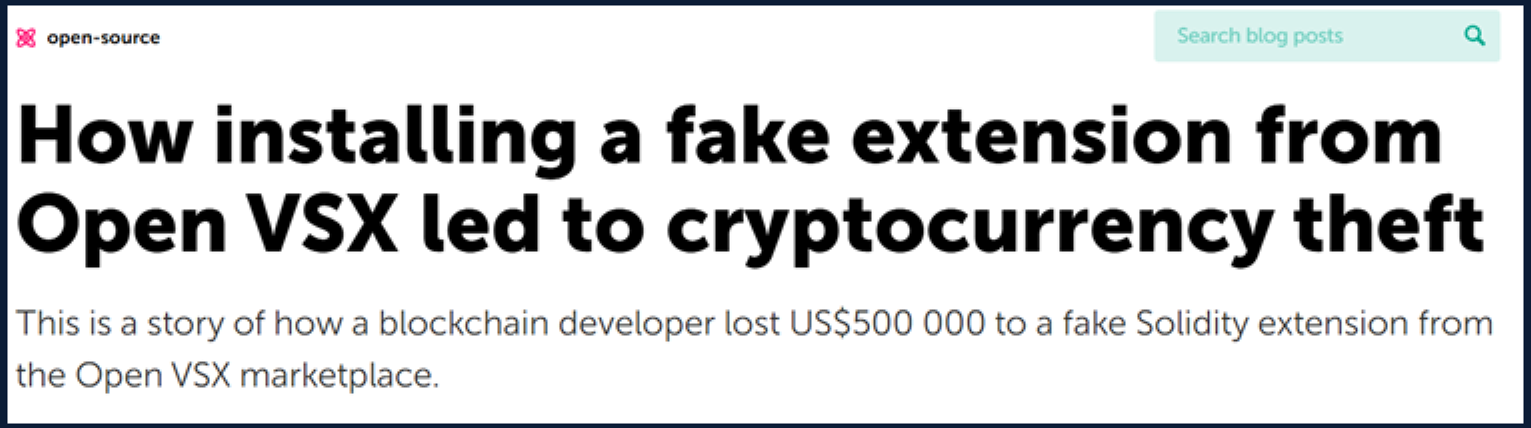
But this not just a VSCode issue

But this isn't just for VSCode...



Extensions for VS Code Compatible Editors

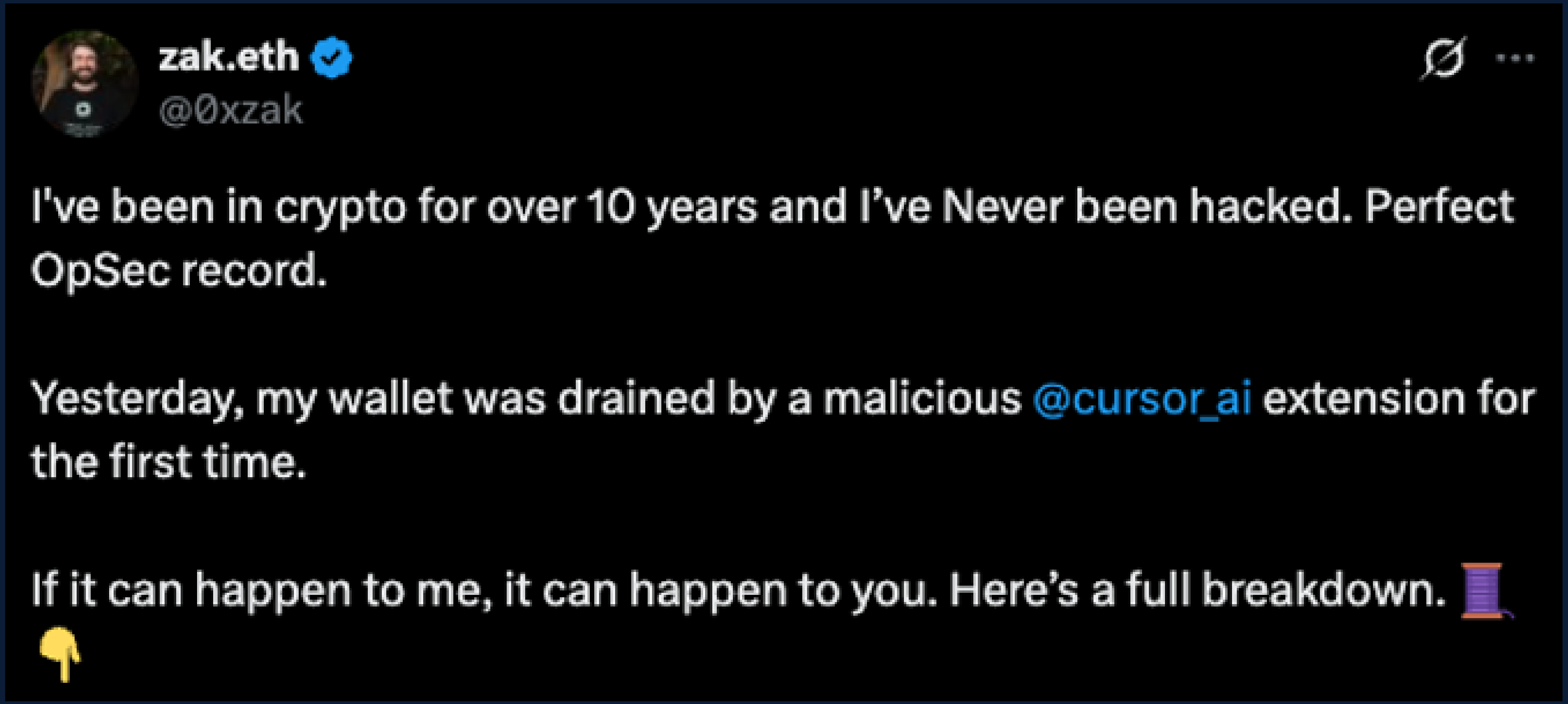
Language Suppo...	redhat	1.45.2025081405	2.9M
GitLens — Git su...	eamodio	2025.8.2105	3M
rust-analyzer	rust-lang	0.4.2588	1.8M
Material Icon Th...	PKief	5.26.0	1.2M
YAML	redhat	1.18.0	1.5M
Continue - open-...	Continue	1.1.79	503K
Flutter	Dart-Code	3.116.2	959K
Dart	Dart-Code	3.116.0	1M
GitLab Workflow	GitLab	6.38.1	531K
clangd	llvm-vs-code-extensi...	0.2...	866K
Cline	saoudrizwan	3.26.1	982K
Code Spell Chec...	streetsidesoftware	4.2.3	690K


open-source

Search blog posts

How installing a fake extension from Open VSX led to cryptocurrency theft

This is a story of how a blockchain developer lost US$500 000 to a fake Solidity extension from the Open VSX marketplace.

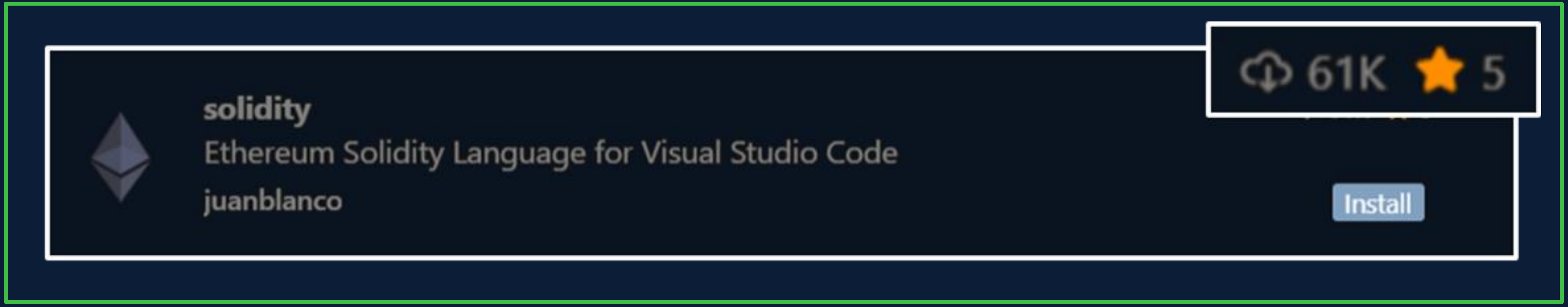

zak.eth
@0xzak

I've been in crypto for over 10 years and I've Never been hacked. Perfect OpSec record.

Yesterday, my wallet was drained by a malicious @cursor_ai extension for the first time.

If it can happen to me, it can happen to you. Here's a full breakdown. 🧵👇

Spot the Malicious Extension

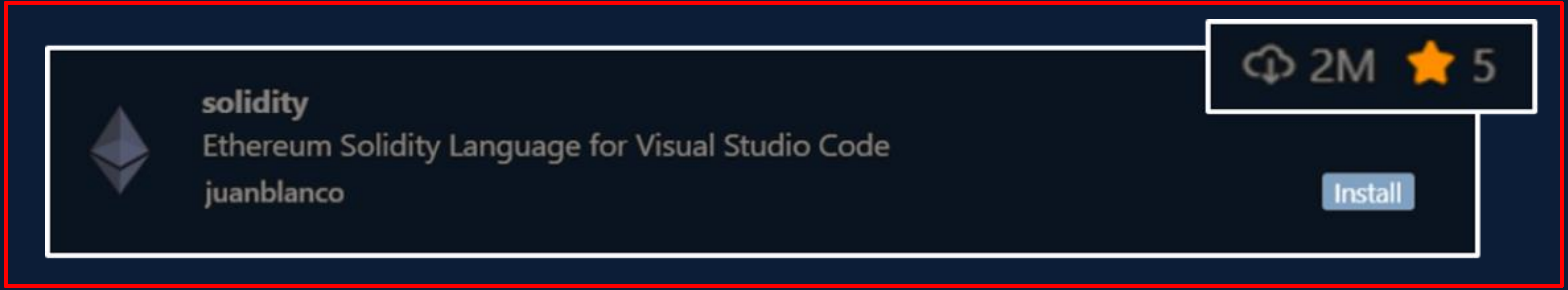


61K 5

solidity

Ethereum Solidity Language for Visual Studio Code

juanblanco



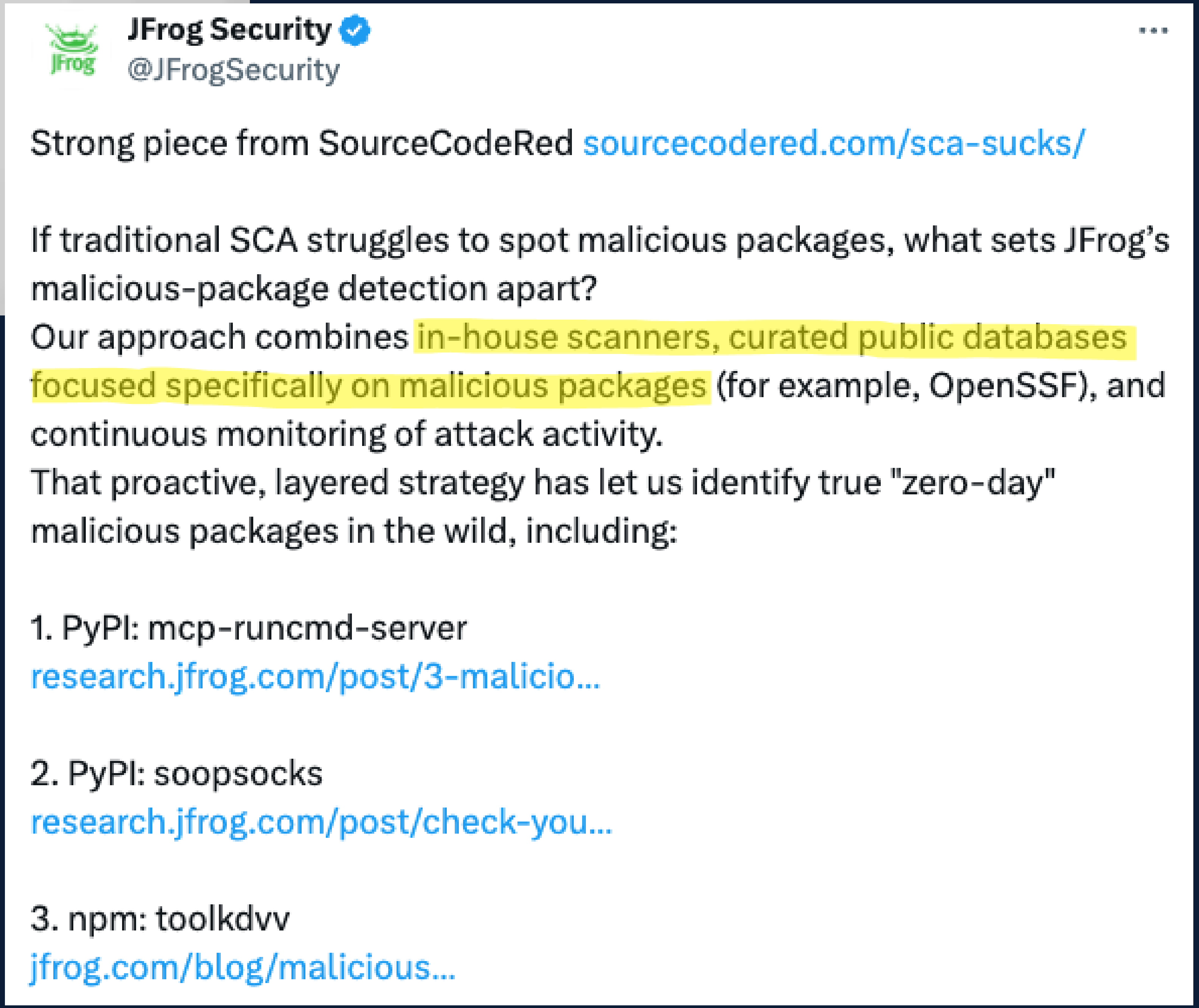
2M 5

solidity

Ethereum Solidity Language for Visual Studio Code

juanblanco

Why SCA Sucks (at detecting malicious packages)



JFrog Security
@JFrogSecurity

Strong piece from SourceCodeRed sourcecodered.com/sca-sucks/

If traditional SCA struggles to spot malicious packages, what sets JFrog's malicious-package detection apart?
Our approach combines in-house scanners, curated public databases focused specifically on malicious packages (for example, OpenSSF), and continuous monitoring of attack activity.
That proactive, layered strategy has let us identify true "zero-day" malicious packages in the wild, including:

1. PyPI: mcp-runcmd-server
research.jfrog.com/post/3-malicio...

2. PyPI: soopsocks
research.jfrog.com/post/check-you...

3. npm: toolkdvv
jfrog.com/blog/malicious...

The Solution

We Need to Curate the Packages used in + the Tools used to **Develop Software**

Prevent The Next Attack With JFrog Curation

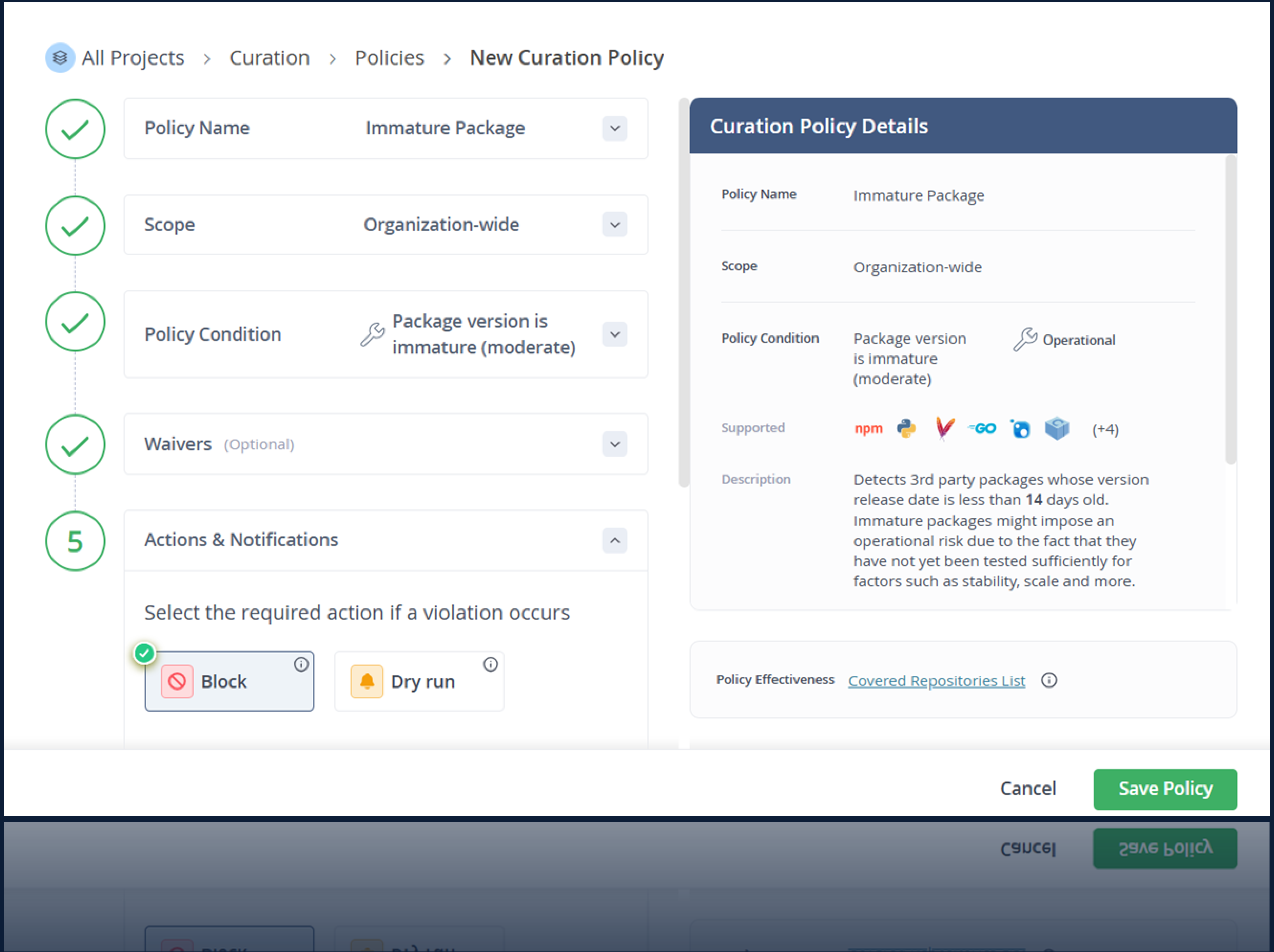
JFrog's security research team investigated the lifespan of hijack attack

- Point Developers to Artifactory
- Set policies to eliminate hijack attack risk

  Block new/immature packages (based on age)

- Make "Compliant Version Selection" effortless

  Divert package managers to mature versions for a seamless dev experience



All Projects > Curation > Policies > New Curation Policy

Policy Name	Immature Package
Scope	Organization-wide
Policy Condition	Package version is immature (moderate)
Waivers (Optional)	

5 Actions & Notifications

Select the required action if a violation occurs

[✓ Block] Dry run

Curation Policy Details

Policy Name	Immature Package
Scope	Organization-wide
Policy Condition	Package version is immature (moderate) — Operational
Supported	npm (+4)
Description	Detects 3rd party packages whose version release date is less than **14** days old. Immature packages might impose an operational risk due to the fact that they have not yet been tested sufficiently for factors such as stability, scale and more.

Policy Effectiveness: Covered Repositories List

Live Audit

12 Month period

~110k New Packages
Downloaded

4
Malicious

1,253
CVEs

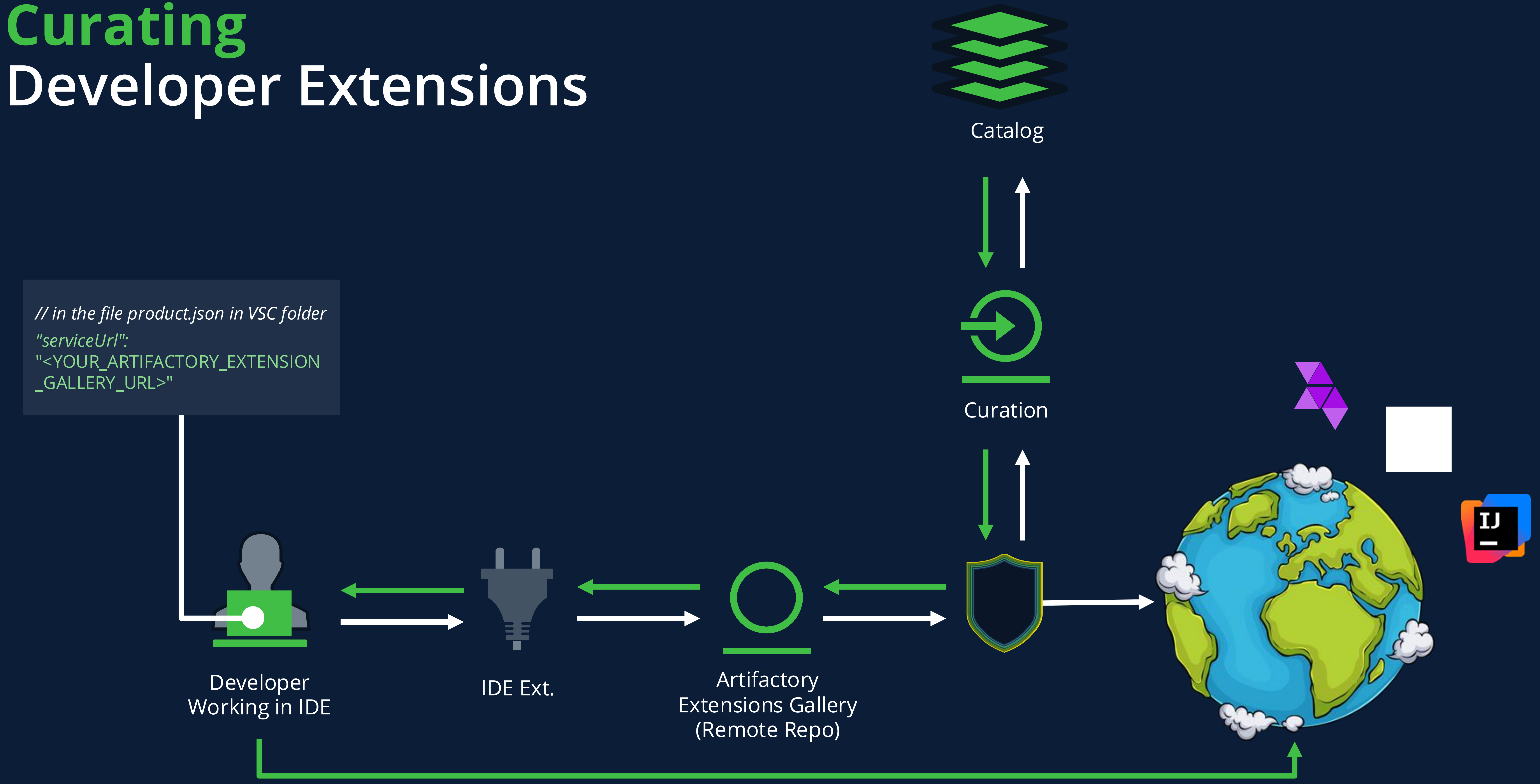
3,803
Licenses violations

1,021
Operational Risks

11,000
hours of remediation saved

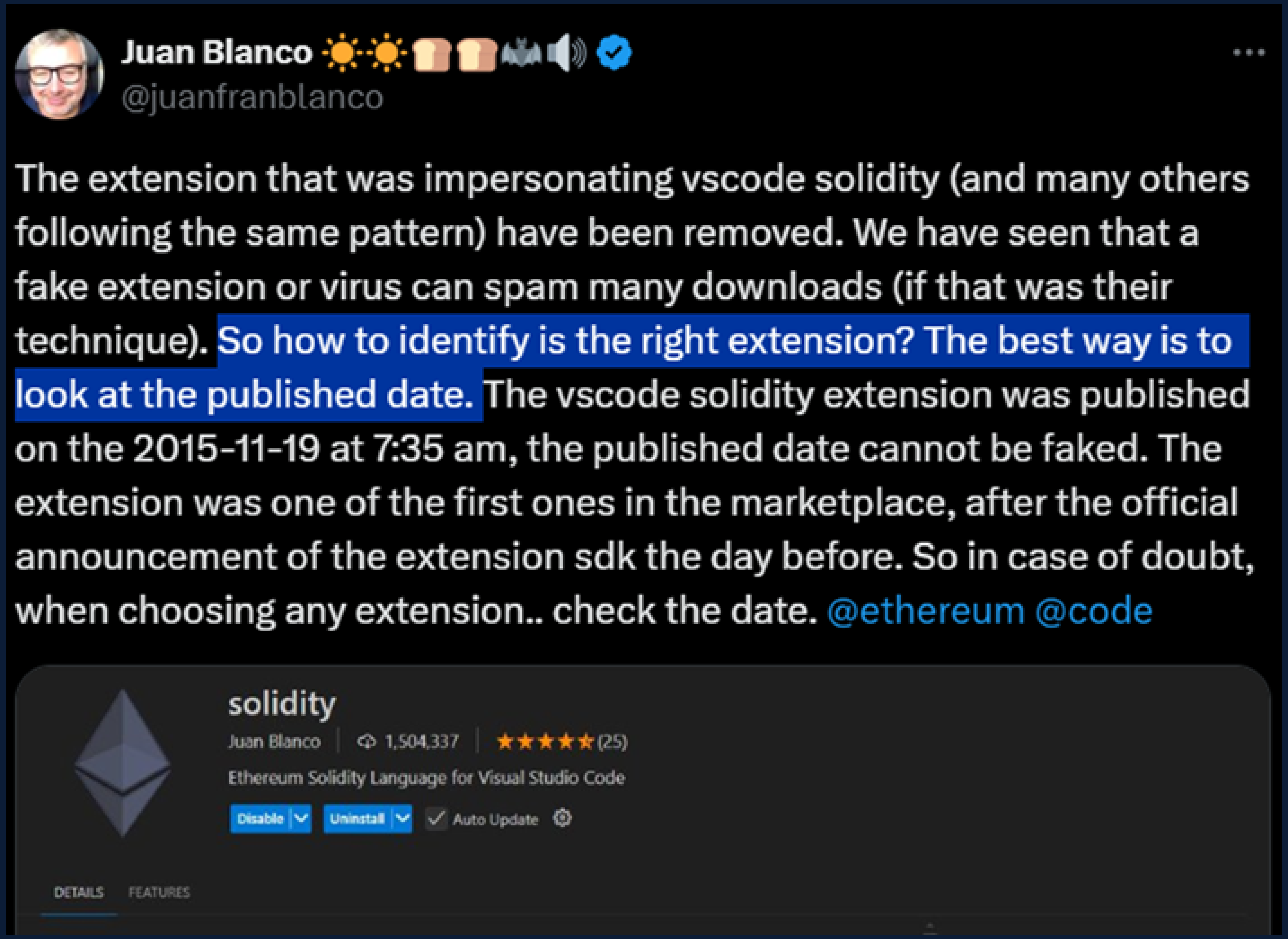
Curation Works.

Curating
Developer Extensions



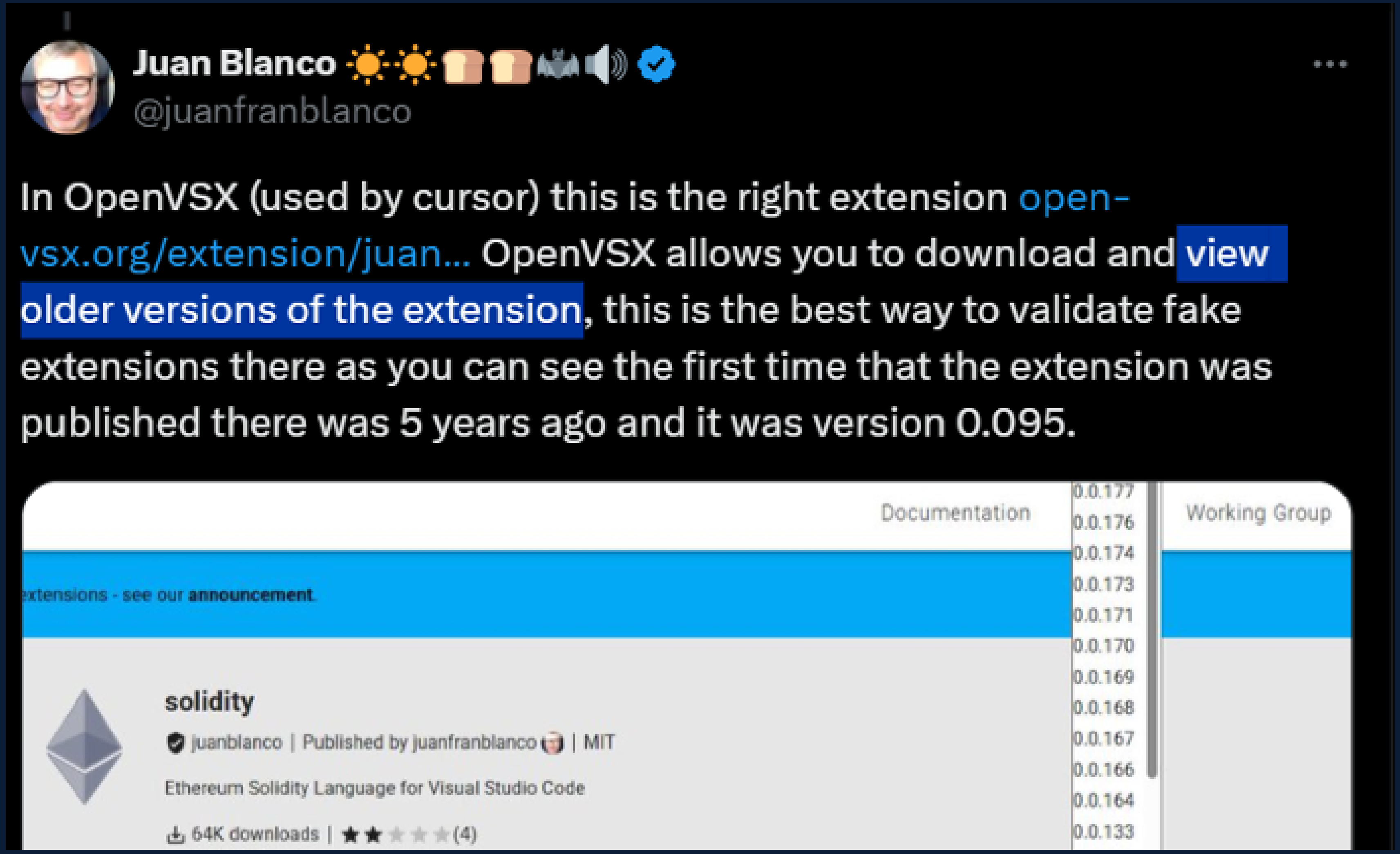
```
// in the file product.json in VSC folder
"serviceUrl":
"<YOUR_ARTIFACTORY_EXTENSION
_GALLERY_URL>"
```

The Challenge


Juan Blanco ☀️☀️🍞🍞🦇🔈
@juanfranblanco

The extension that was impersonating vscode solidity (and many others following the same pattern) have been removed. We have seen that a fake extension or virus can spam many downloads (if that was their technique). So how to identify is the right extension? The best way is to look at the published date. The vscode solidity extension was published on the 2015-11-19 at 7:35 am, the published date cannot be faked. The extension was one of the first ones in the marketplace, after the official announcement of the extension sdk the day before. So in case of doubt, when choosing any extension.. check the date. @ethereum @code

solidity
Juan Blanco | 1,504,337 | ★★★★★ (25)
Ethereum Solidity Language for Visual Studio Code

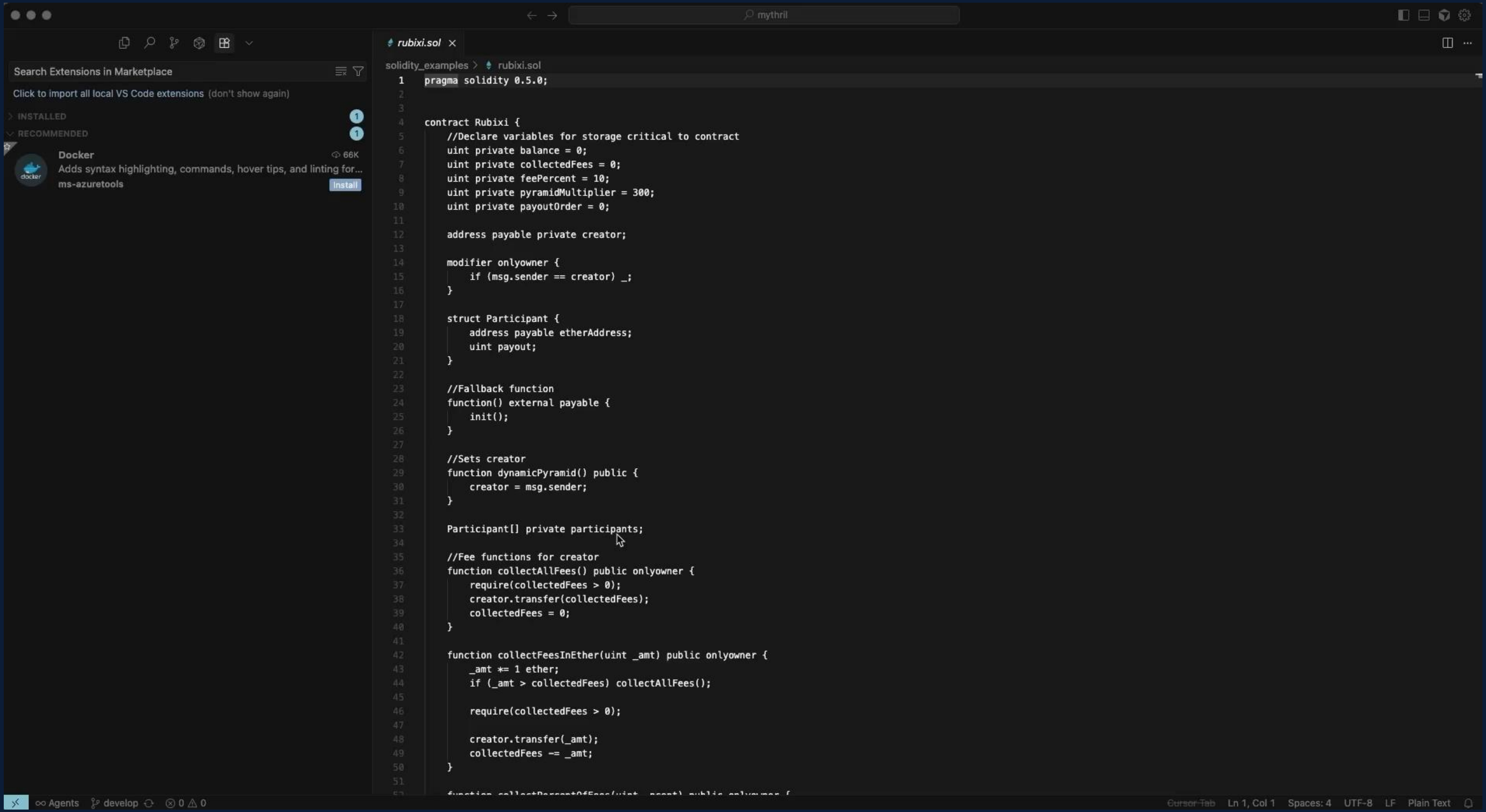

Juan Blanco ☀️☀️🍞🍞🦇🔈
@juanfranblanco

In OpenVSX (used by cursor) this is the right extension open-vsx.org/extension/juan... OpenVSX allows you to download and view older versions of the extension, this is the best way to validate fake extensions there as you can see the first time that the extension was published there was 5 years ago and it was version 0.095.

solidity
juanblanco | Published by juanfranblanco | MIT
Ethereum Solidity Language for Visual Studio Code
64K downloads | ★★ (4)

0.0.177
0.0.176
0.0.174
0.0.173
0.0.171
0.0.170
0.0.169
0.0.168
0.0.167
0.0.166
0.0.164
0.0.133

Developer Extension Security Scope

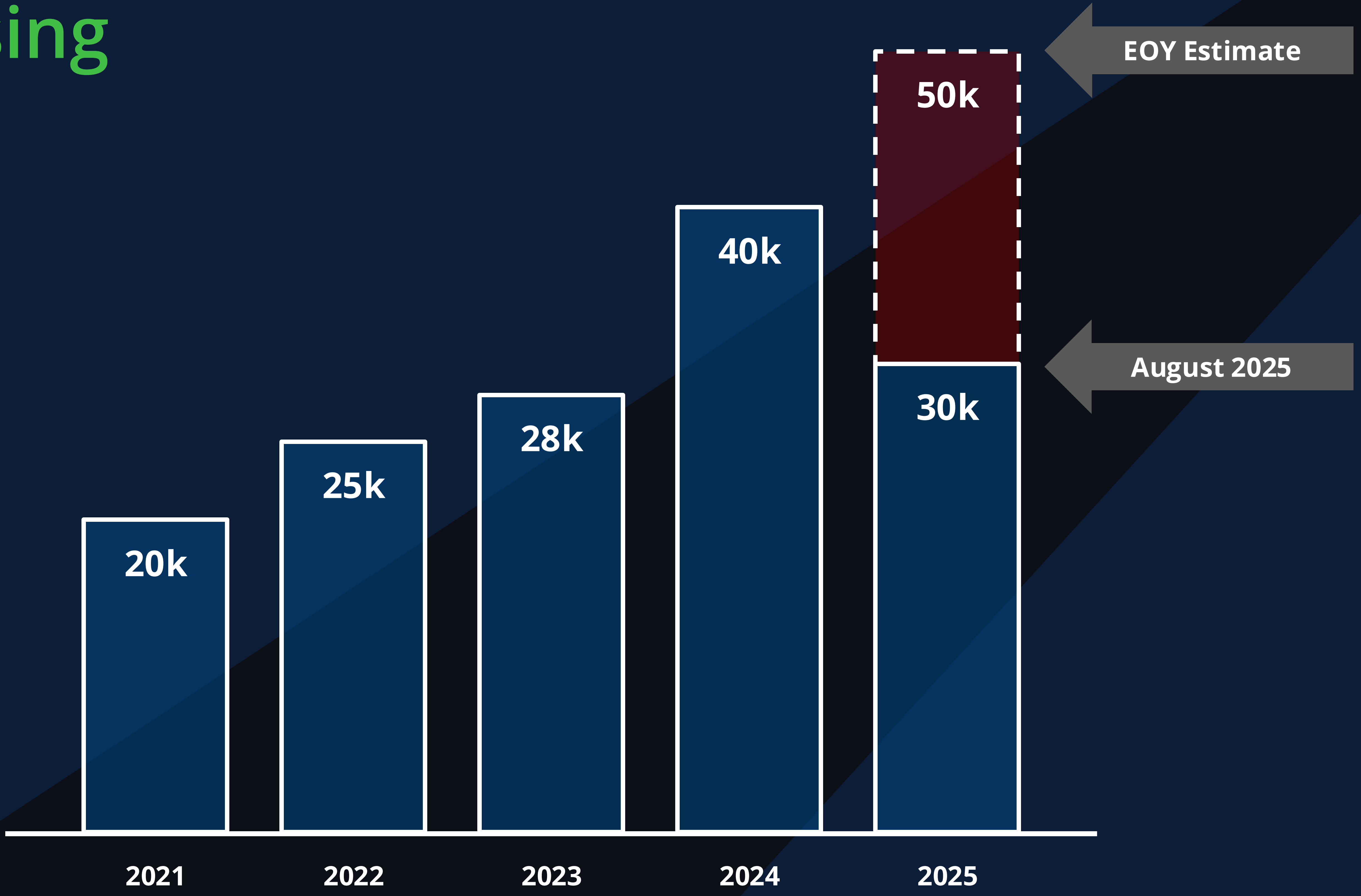

mythril
Search Extensions in Marketplace
Click to import all local VS Code extensions (don't show again)
INSTALLED
RECOMMENDED
Docker
66K
Adds syntax highlighting, commands, hover tips, and linting for...
ms-azuretools
Install
rubixi.sol
solidity_examples > rubixi.sol
Agents
develop
Cursor Tab
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Plain Text

```
pragma solidity 0.5.0;


contract Rubixi {
    //Declare variables for storage critical to contract
    uint private balance = 0;
    uint private collectedFees = 0;
    uint private feePercent = 10;
    uint private pyramidMultiplier = 300;
    uint private payoutOrder = 0;

    address payable private creator;

    modifier onlyowner {
        if (msg.sender == creator) _;
    }

    struct Participant {
        address payable etherAddress;
        uint payout;
    }

    //Fallback function
    function() external payable {
        init();
    }

    //Sets creator
    function dynamicPyramid() public {
        creator = msg.sender;
    }

    Participant[] private participants;

    //Fee functions for creator
    function collectAllFees() public onlyowner {
        require(collectedFees > 0);
        creator.transfer(collectedFees);
        collectedFees = 0;
    }

    function collectFeesInEther(uint _amt) public onlyowner {
        _amt *= 1 ether;
        if (_amt > collectedFees) collectAllFees();

        require(collectedFees > 0);

        creator.transfer(_amt);
        collectedFees -= _amt;
    }

    function collectPercentOfFees(uint _pcent) public onlyowner {
```

While attackers are targeting the new

Developers are Swamped by the old

Number of CVEs are Ever Increasing

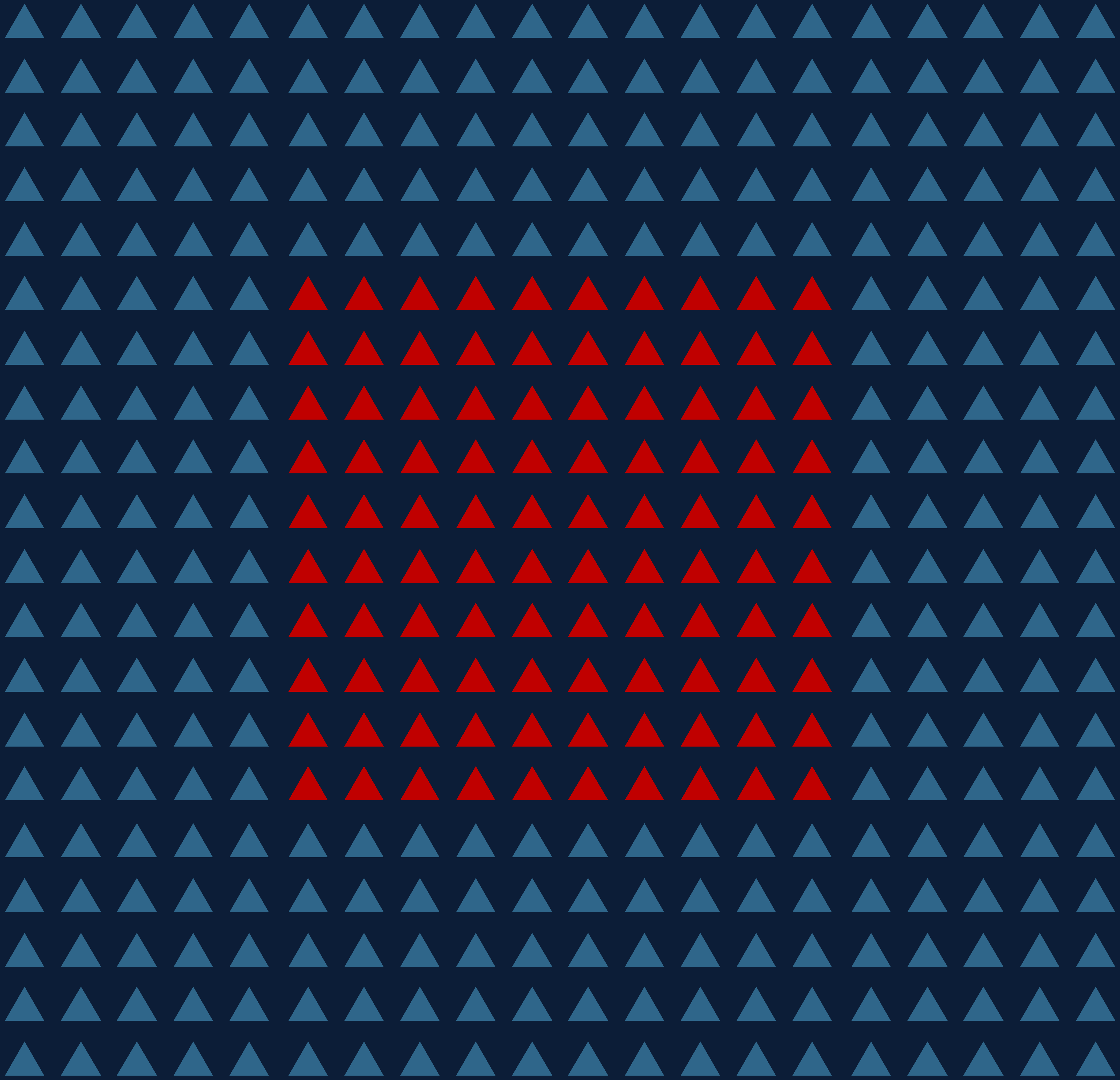


NIST NATIONAL VULNERABILITY DATABASE NVD

SLA times for Fixing CVEs are Constantly Decreasing

We need to focus on

CVEs that are Critical
that Can be Exploited
and Running in Production



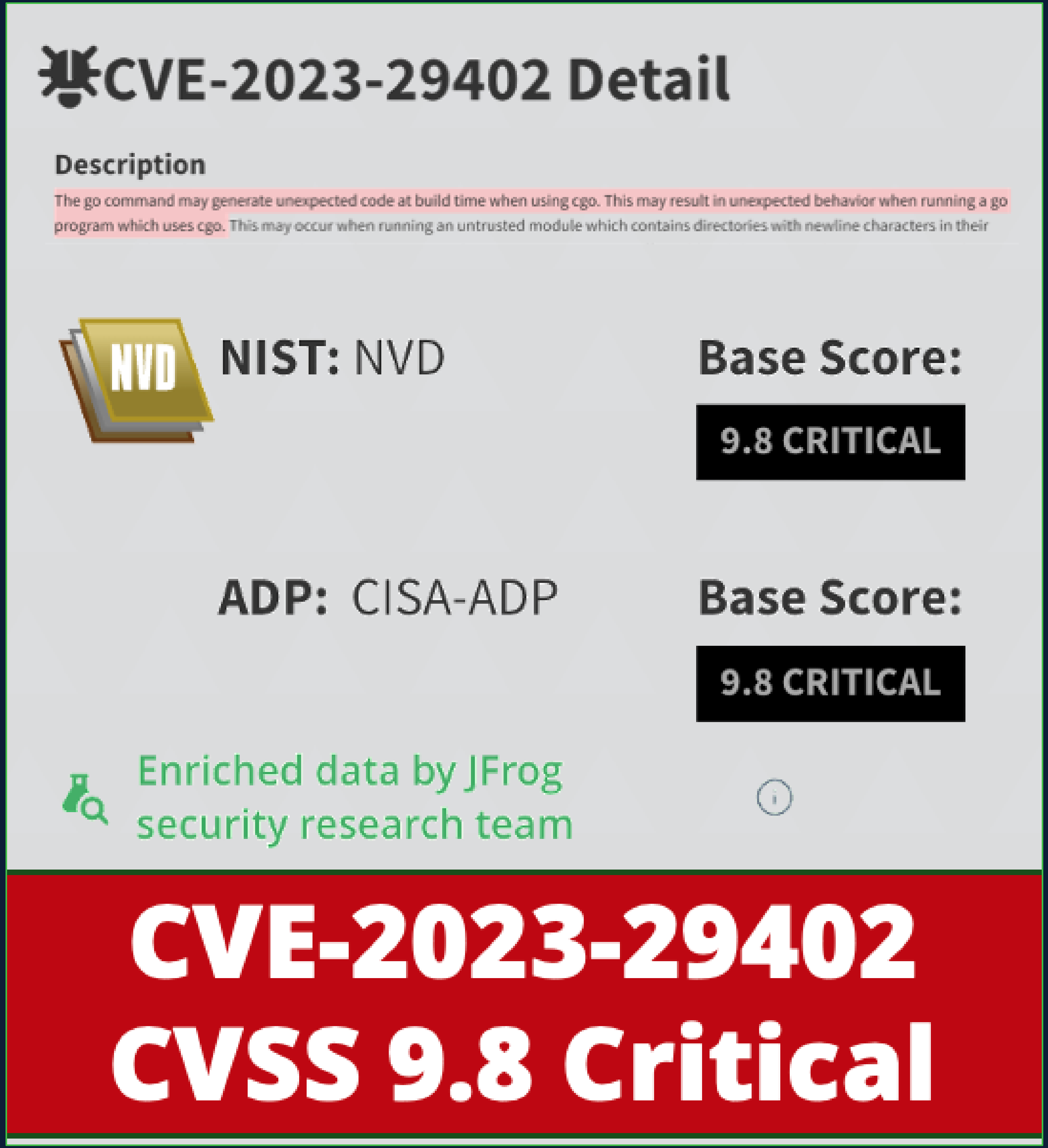
88%

of critical severity CVEs
are grossly inflated

Exploitation Requires the Victim to **Download and Execute an Untrusted Malicious Package**

1 Out of 3

JFrog Customers



CVE-2023-29402 Detail

Description

The go command may generate unexpected code at build time when using cgo. This may result in unexpected behavior when running a go program which uses cgo. This may occur when running an untrusted module which contains directories with newline characters in their

NIST: NVD	**Base Score:** 9.8 CRITICAL
ADP: CISA-ADP	**Base Score:** 9.8 CRITICAL

Enriched data by JFrog security research team

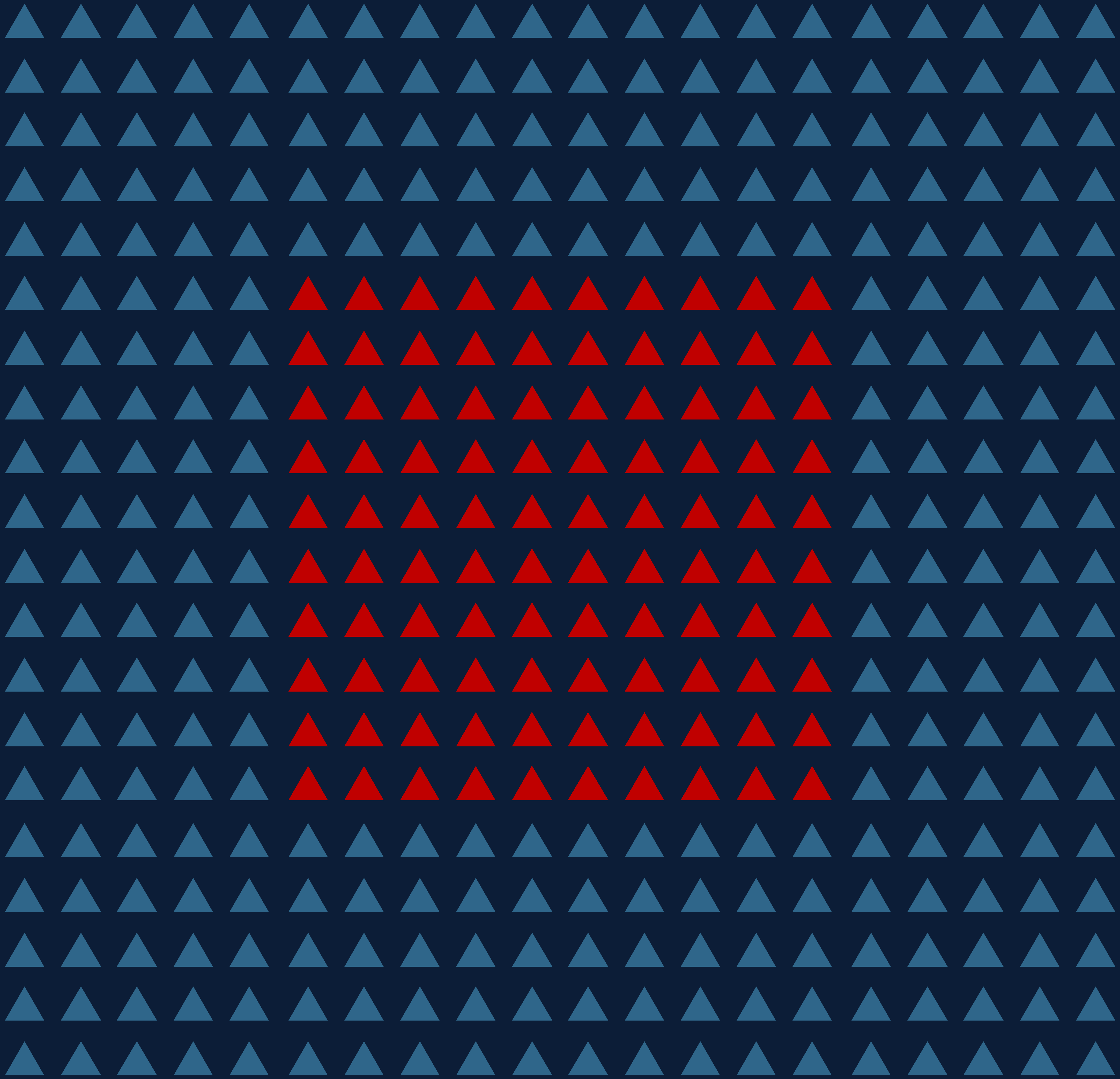
CVE-2023-29402
CVSS 9.8 Critical

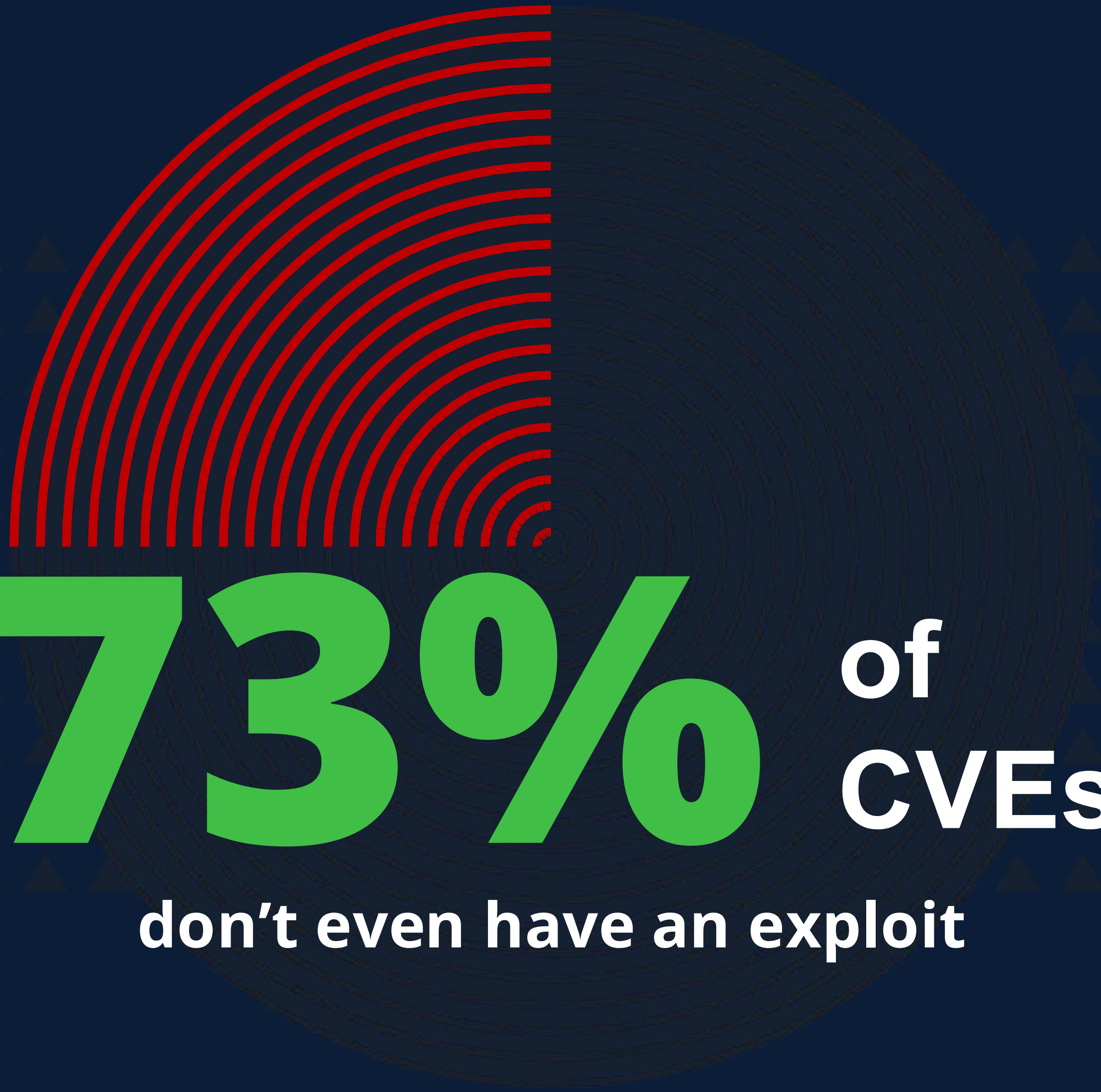
We need to focus on

CVEs that are ~~Critical~~

that Can be Exploited

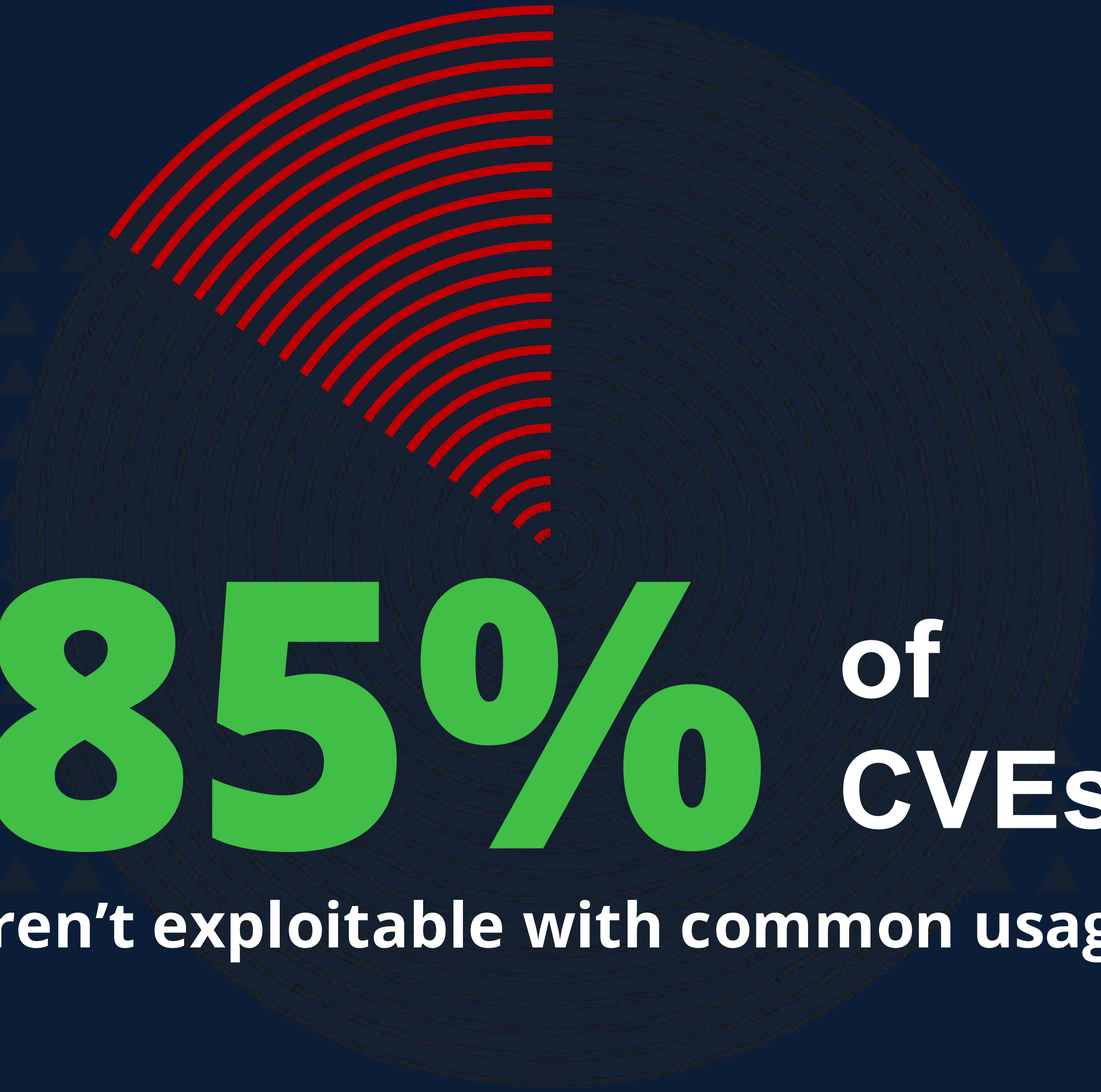
and Running in Production





73% of CVEs

don’t even have an exploit



85% of CVEs

aren’t exploitable with common usage

CVE Contextual Analysis

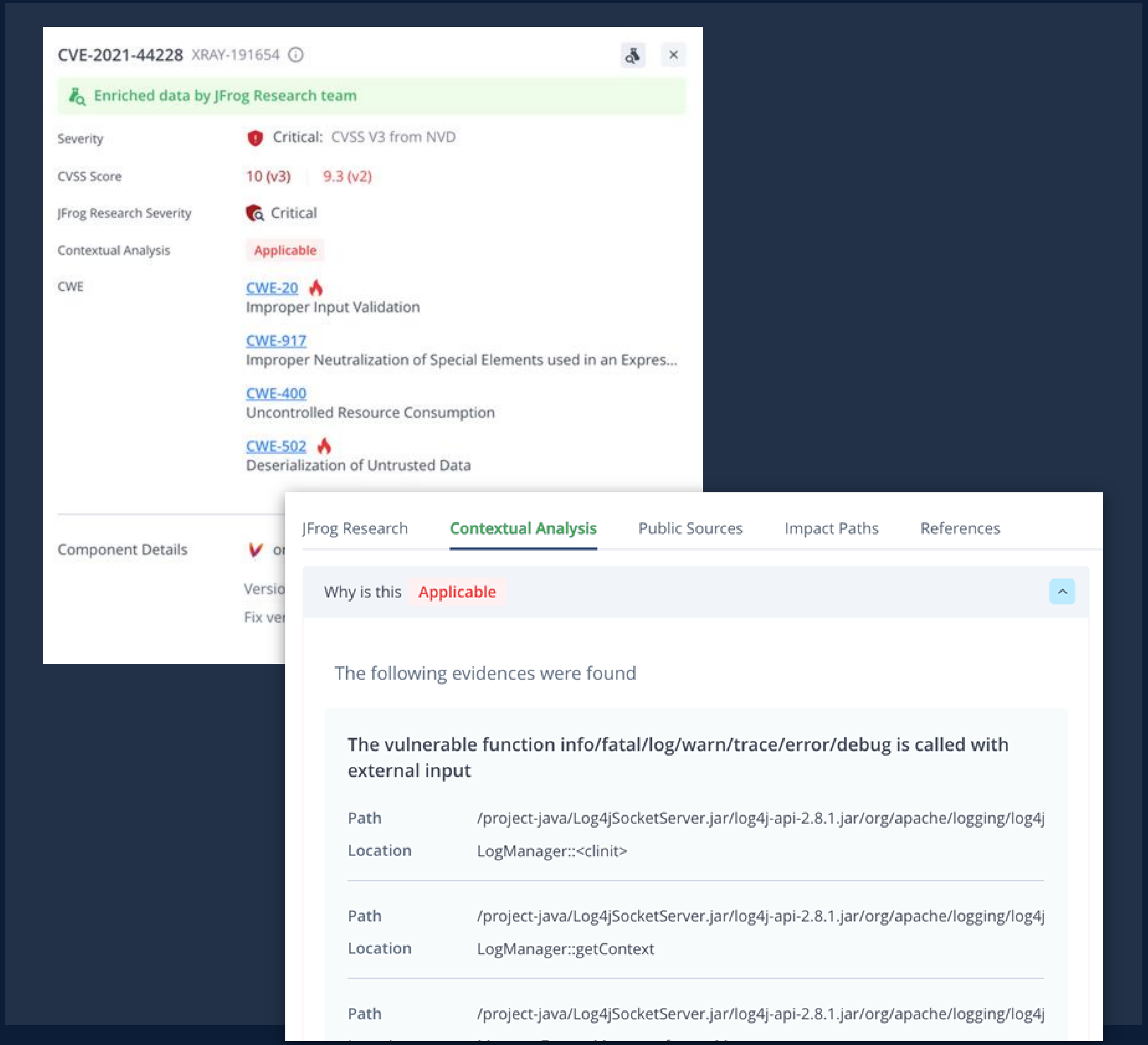
WHAT DOES IT DO?

- Flags vulnerabilities that are **applicable** and **exploitable** in your organization
- Provides clear and **research-backed remediation guidance**

WHY IS IT IMPORTANT?

- Eliminates noise by focusing on **relevant vulnerabilities only**
- Improves **remediation efficiency**, eliminates wasted effort

88% Critical and 57% High CVE scores are not as severe as indicated by the CVSS score



CVE-2021-44228 XRAY-191654

Enriched data by JFrog Research team

Severity	Critical: CVSS V3 from NVD
CVSS Score	10 (v3) | 9.3 (v2)
JFrog Research Severity	Critical
Contextual Analysis	Applicable
CWE	CWE-20 Improper Input Validation; CWE-917 Improper Neutralization of Special Elements used in an Expres...; CWE-400 Uncontrolled Resource Consumption; CWE-502 Deserialization of Untrusted Data
Component Details	

JFrog Research | Contextual Analysis | Public Sources | Impact Paths | References

Why is this Applicable

The following evidences were found

The vulnerable function info/fatal/log/warn/trace/error/debug is called with external input

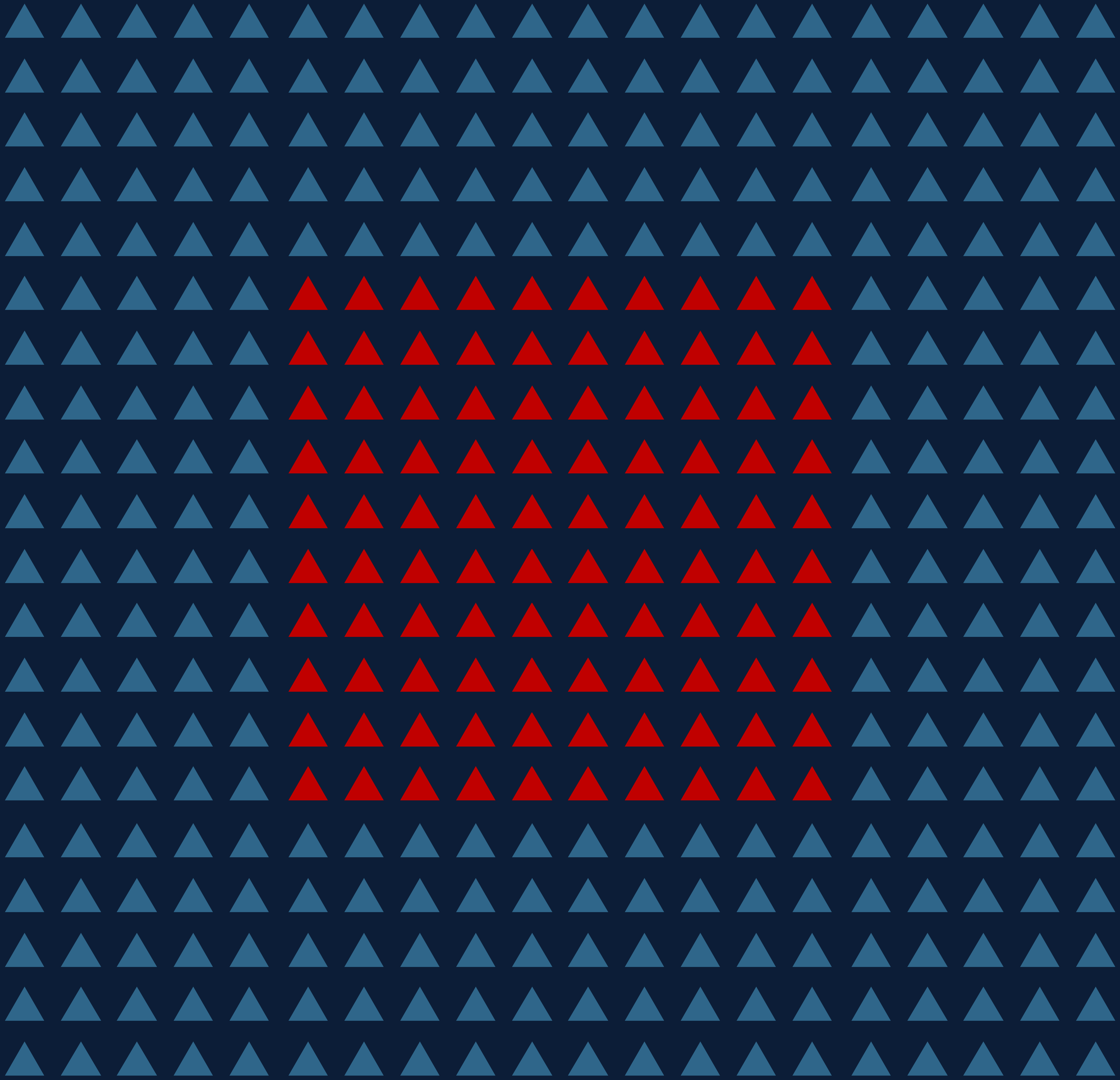
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j
Location	LogManager::<clinit>
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j
Location	LogManager::getContext
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j

We need to focus on

CVEs that are ~~Critical~~

that Can be ~~Exploited~~

and Running in Production



JFrog Runtime

Protect Applications in Runtime

Complete Runtime Visibility

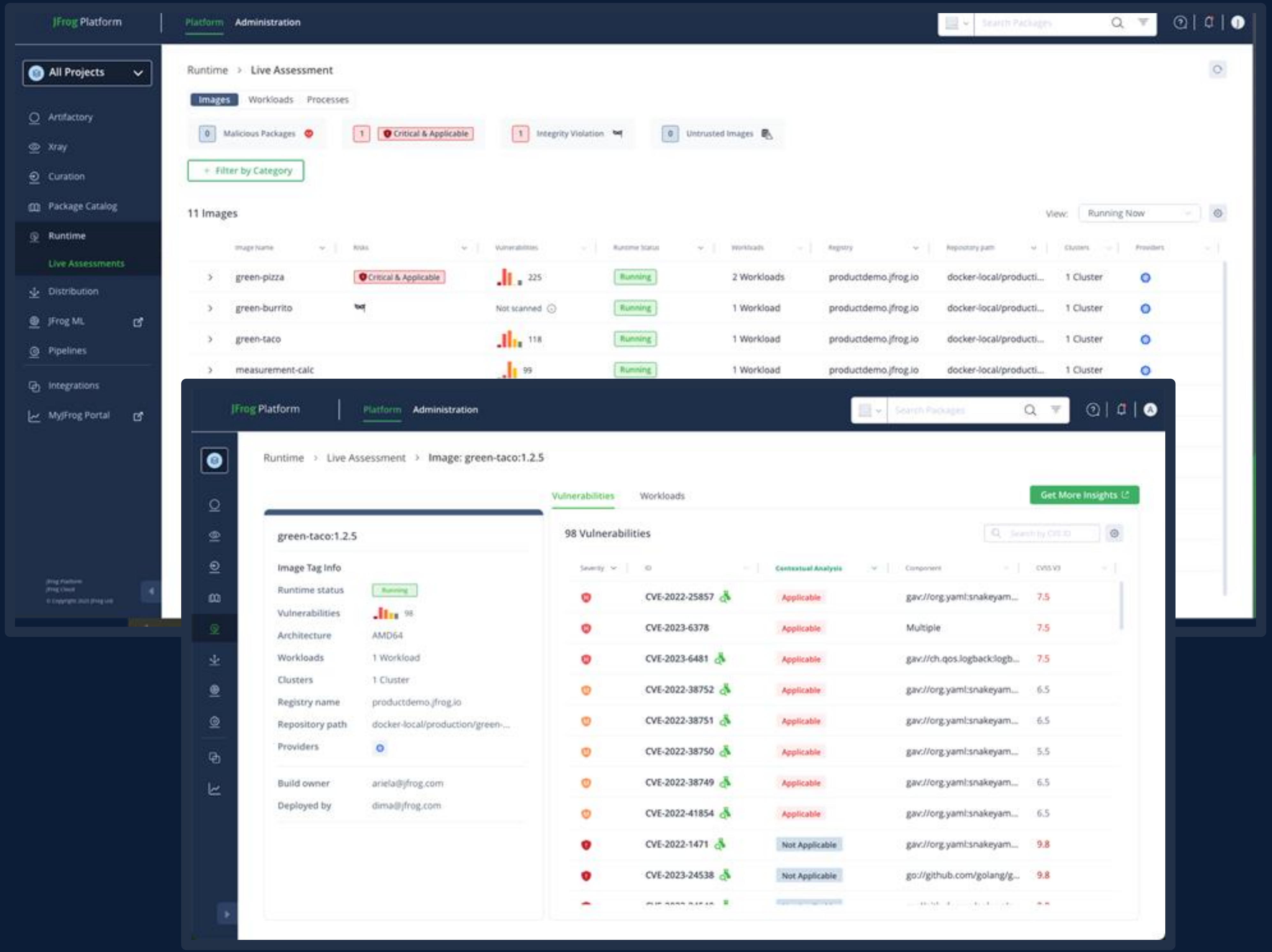
Gain a clear and contextualized view of all running applications and workloads. Ensure Runtime integrity

Accelerated Incident Response and Prioritization

Full visibility to image ownership and deployment history

Automated Security Enforcement & Integrity Checking

Automatically trigger security scans on active images and verify application integrity



Runtime > Live Assessment

0 Malicious Packages · 1 Critical & Applicable · 1 Integrity Violation · 0 Untrusted Images

11 Images

	Image Name	Risks	Vulnerabilities	Runtime Status	Workloads	Registry	Repository path	Clusters	Providers
›	green-pizza	Critical & Applicable	225	Running	2 Workloads	productdemo.jfrog.io	docker-local/producti...	1 Cluster	
›	green-burrito		Not scanned	Running	1 Workload	productdemo.jfrog.io	docker-local/producti...	1 Cluster	
›	green-taco		118	Running	1 Workload	productdemo.jfrog.io	docker-local/producti...	1 Cluster	
›	measurement-calc		99	Running	1 Workload	productdemo.jfrog.io	docker-local/producti...	1 Cluster	

Runtime > Live Assessment > Image: green-taco:1.2.5

green-taco:1.2.5

Image Tag Info

Runtime status	Running
Vulnerabilities	98
Architecture	AMD64
Workloads	1 Workload
Clusters	1 Cluster
Registry name	productdemo.jfrog.io
Repository path	docker-local/production/green-...
Providers	
Build owner	ariela@jfrog.com
Deployed by	dima@jfrog.com

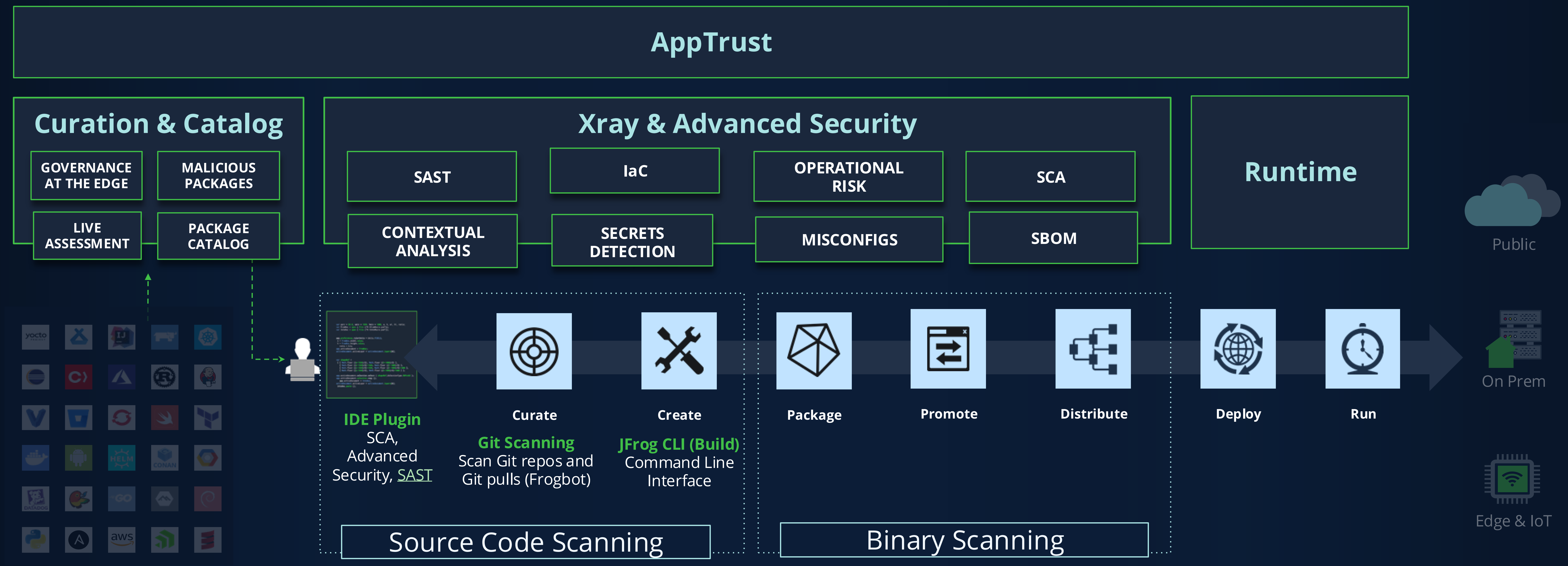
98 Vulnerabilities

Severity	ID	Contextual Analysis	Component	CVSS V3
	CVE-2022-25857	Applicable	gav://org.yaml:snakeyam...	7.5
	CVE-2023-6378	Applicable	Multiple	7.5
	CVE-2023-6481	Applicable	gav://ch.qos.logback:logb...	7.5
	CVE-2022-38752	Applicable	gav://org.yaml:snakeyam...	6.5
	CVE-2022-38751	Applicable	gav://org.yaml:snakeyam...	6.5
	CVE-2022-38750	Applicable	gav://org.yaml:snakeyam...	5.5
	CVE-2022-38749	Applicable	gav://org.yaml:snakeyam...	6.5
	CVE-2022-41854	Applicable	gav://org.yaml:snakeyam...	6.5
	CVE-2022-1471	Not Applicable	gav://org.yaml:snakeyam...	9.8
	CVE-2023-24538	Not Applicable	go://github.com/golang/g...	9.8

We need to focus on

CVEs that are ~~Critical~~
that can be ~~Exploited~~
and Running in ~~Production~~

End-to-end Security with JFrog



JFrog

Thank You